Eliminating Popular Faces in Curve Arrangements

21.09.2023 ◆ GD 2023 ◆ Session 6

TU WIEN ac ALGORITHMS AND COMPLEXITY GROUP

Soeren Terziadis

Utrecht University

Phoebe de Nooijer
Tamara Mchedlidze
Maarten Löffler

TU Graz

Alexandra Weinberger

IST AUSTRIA

Zuzana Masárová

Freie Universität Berlin

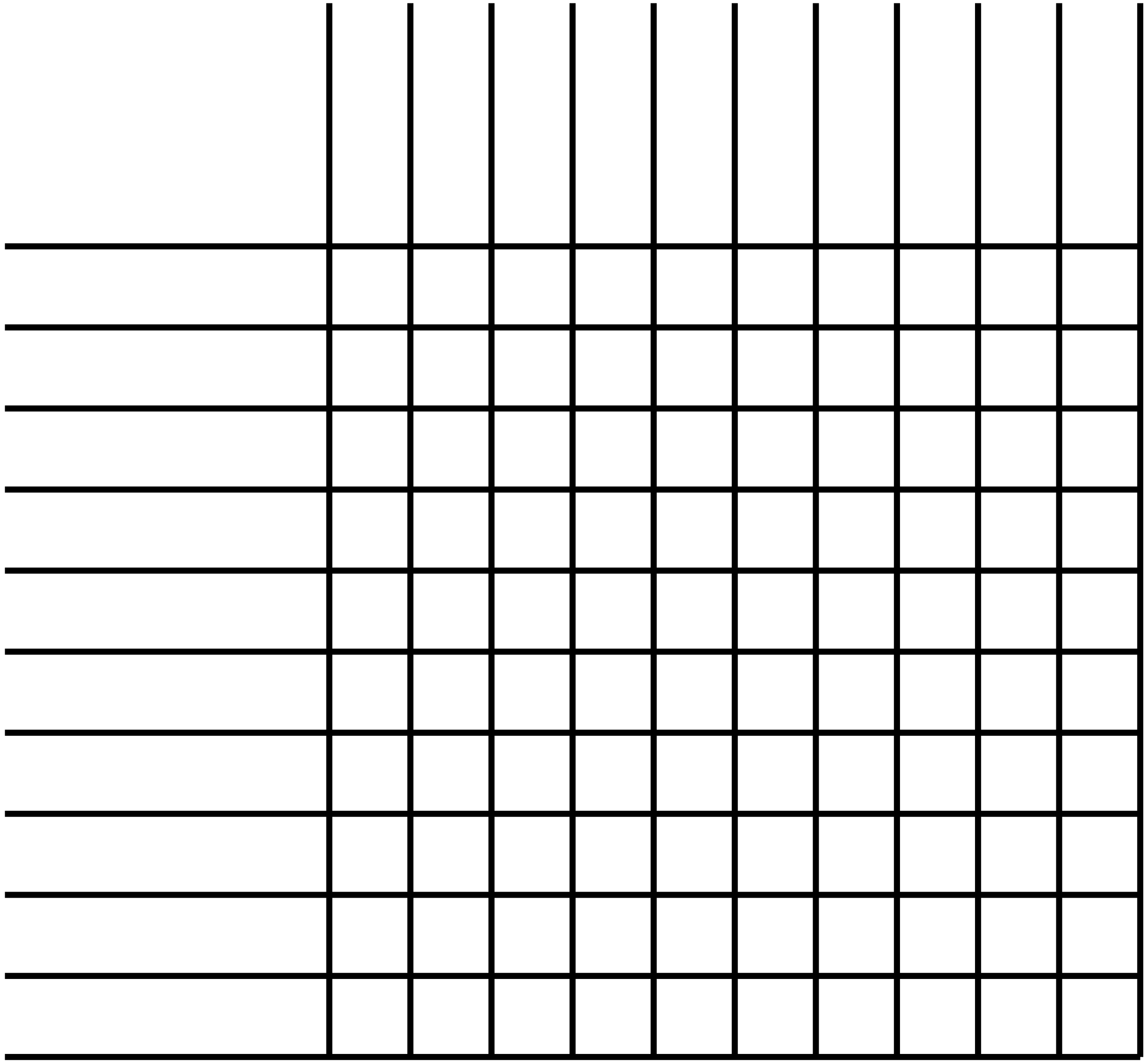
Günter Rote

A project started at 16th European Research Week on Geometric Graphs (GGWeek) in Strobl (AT), 2019

Outline

Nonograms

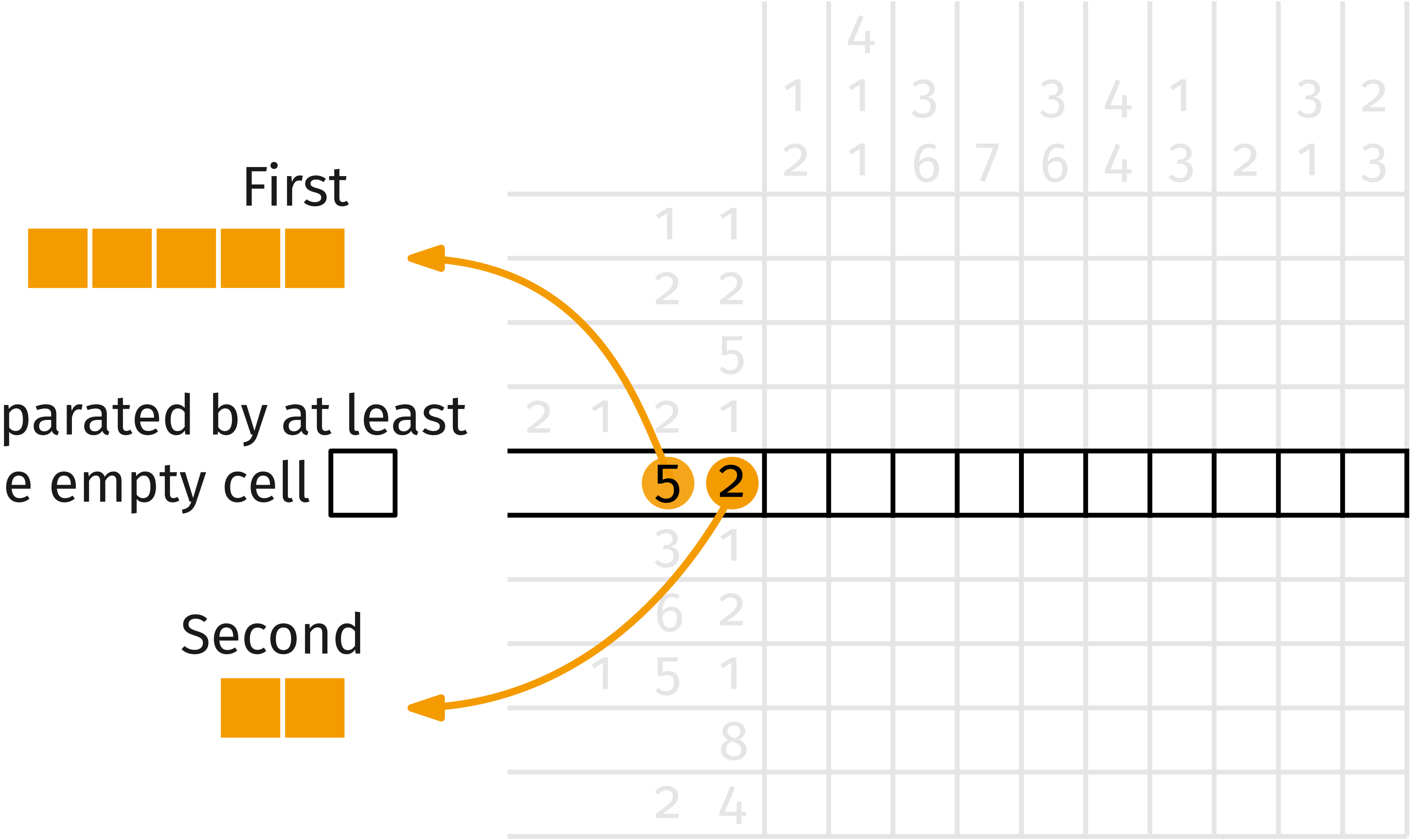
Nonograms (Griddlers, 判じ絵, Picross,...)



Nonograms (Griddlers, 判じ絵, Picross,...)

	1 2	4 1 1	3 6	7	3 6	4 4	1 3	2	3 1	2 3
1 1										
2 2										
5										
2 1 2 1										
5 2										
3 1										
6 2										
1 5 1										
8										
2 4										

Nonograms (Griddlers, 判じ絵, Picross,...)

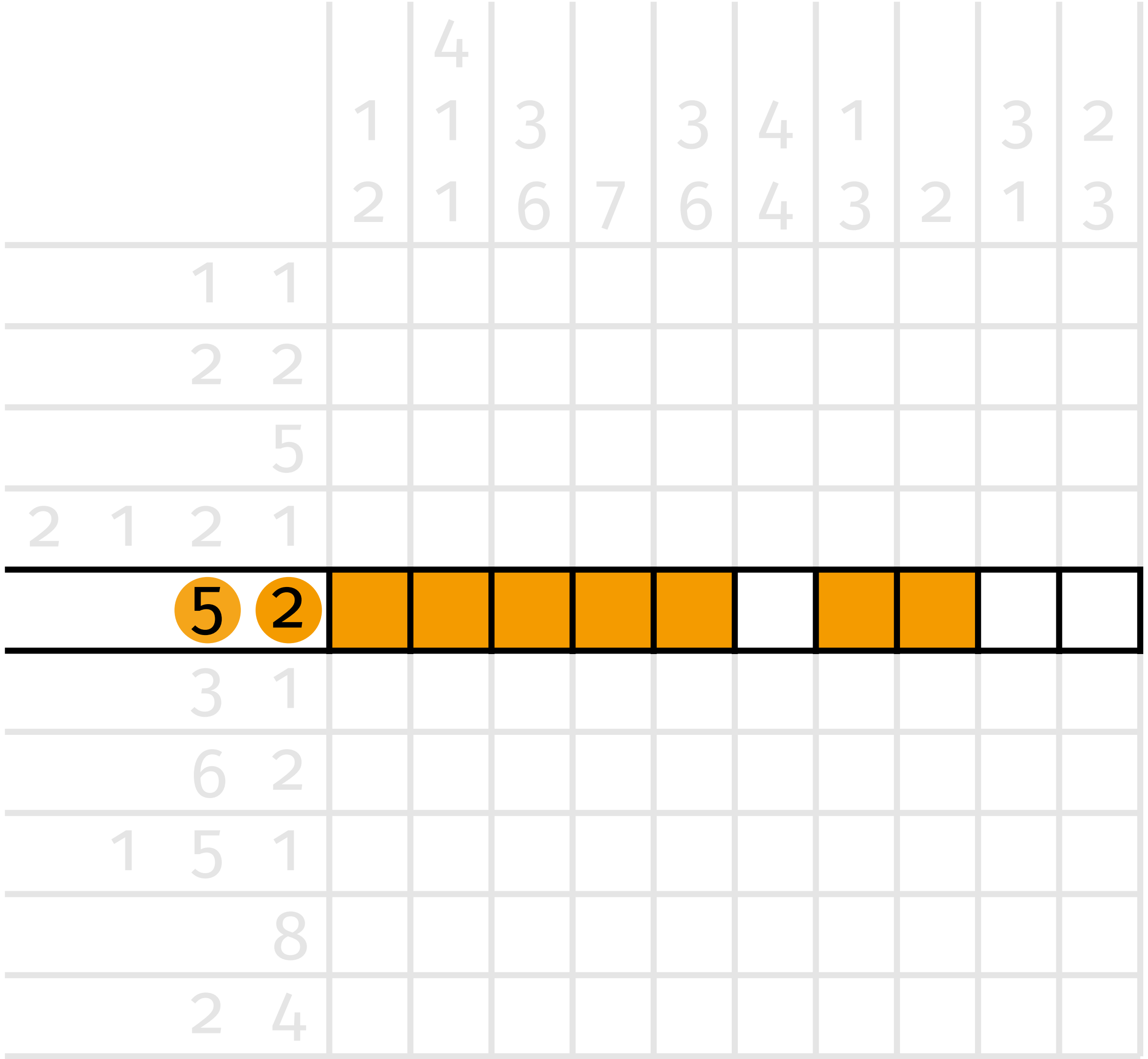


First

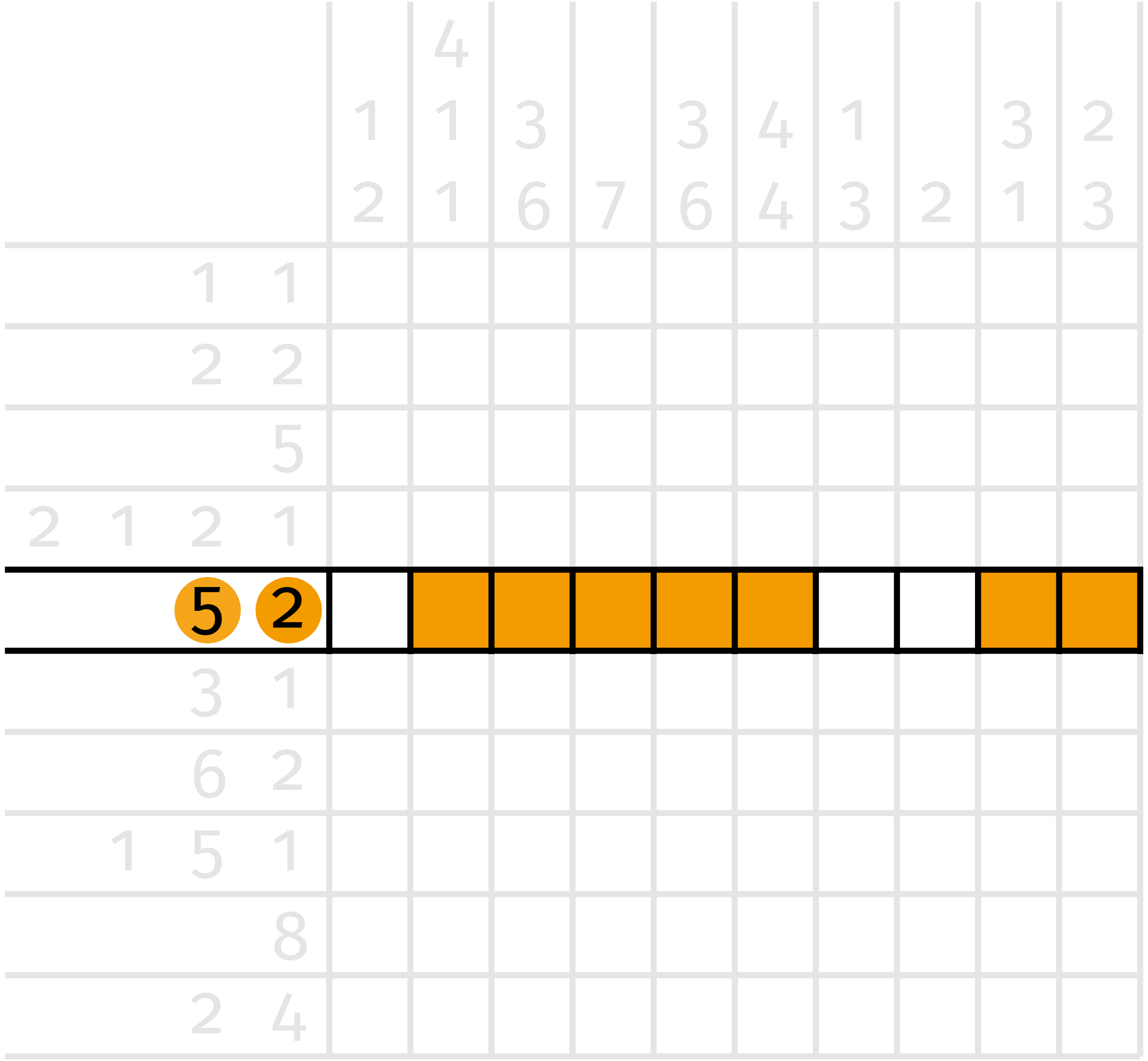
Separated by at least one empty cell ☐

Second

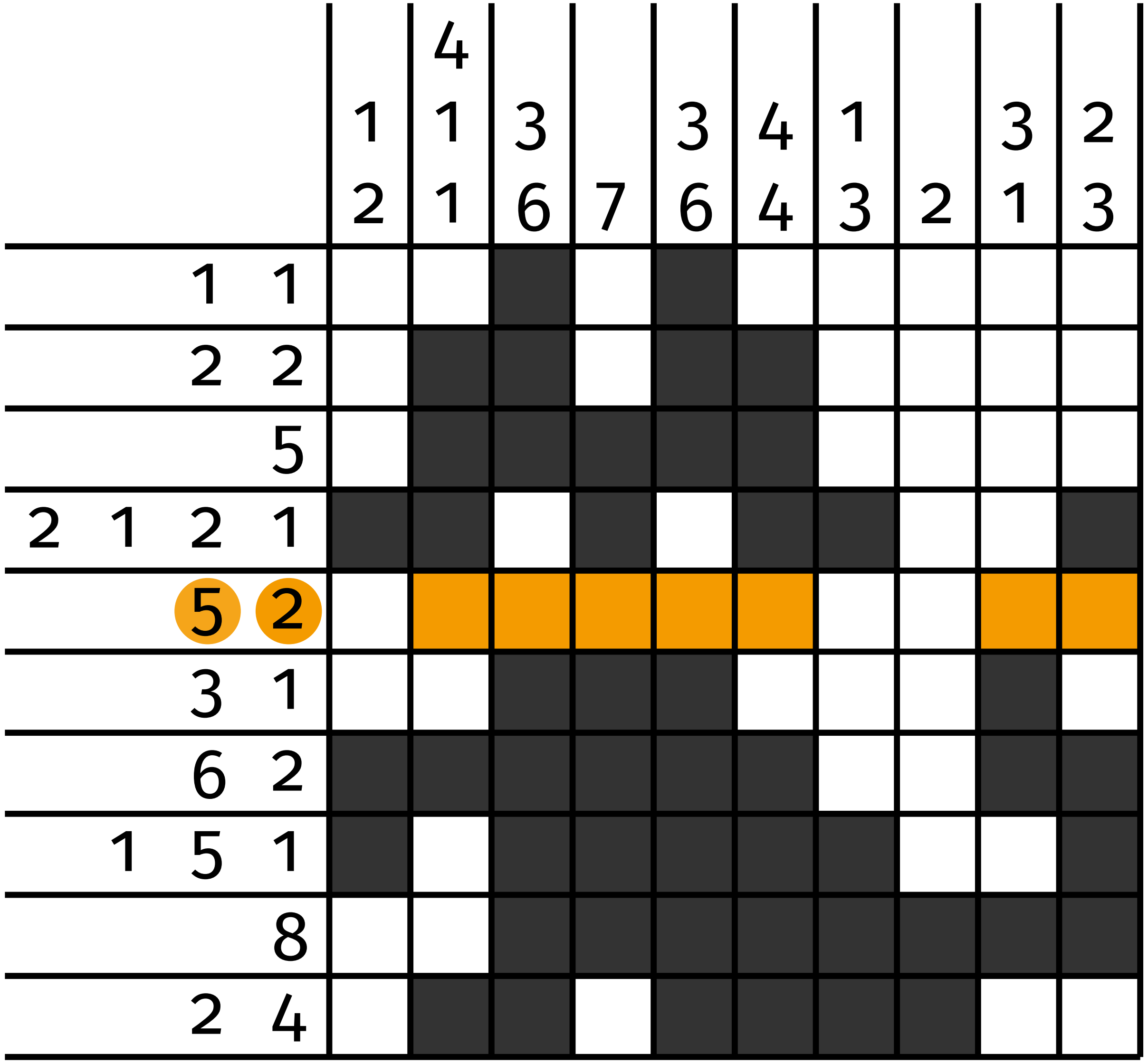
Nonograms (Griddlers, 判じ絵, Picross,...)



Nonograms (Griddlers, 判じ絵, Picross,...)



Nonograms (Griddlers, 判じ絵, Picross,...)

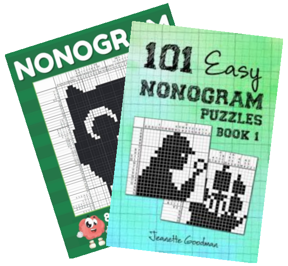


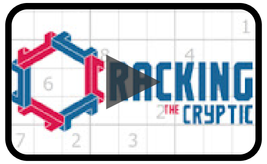
Nonograms (Griddlers, 判じ絵, Picross,...)

Popular with consumers...

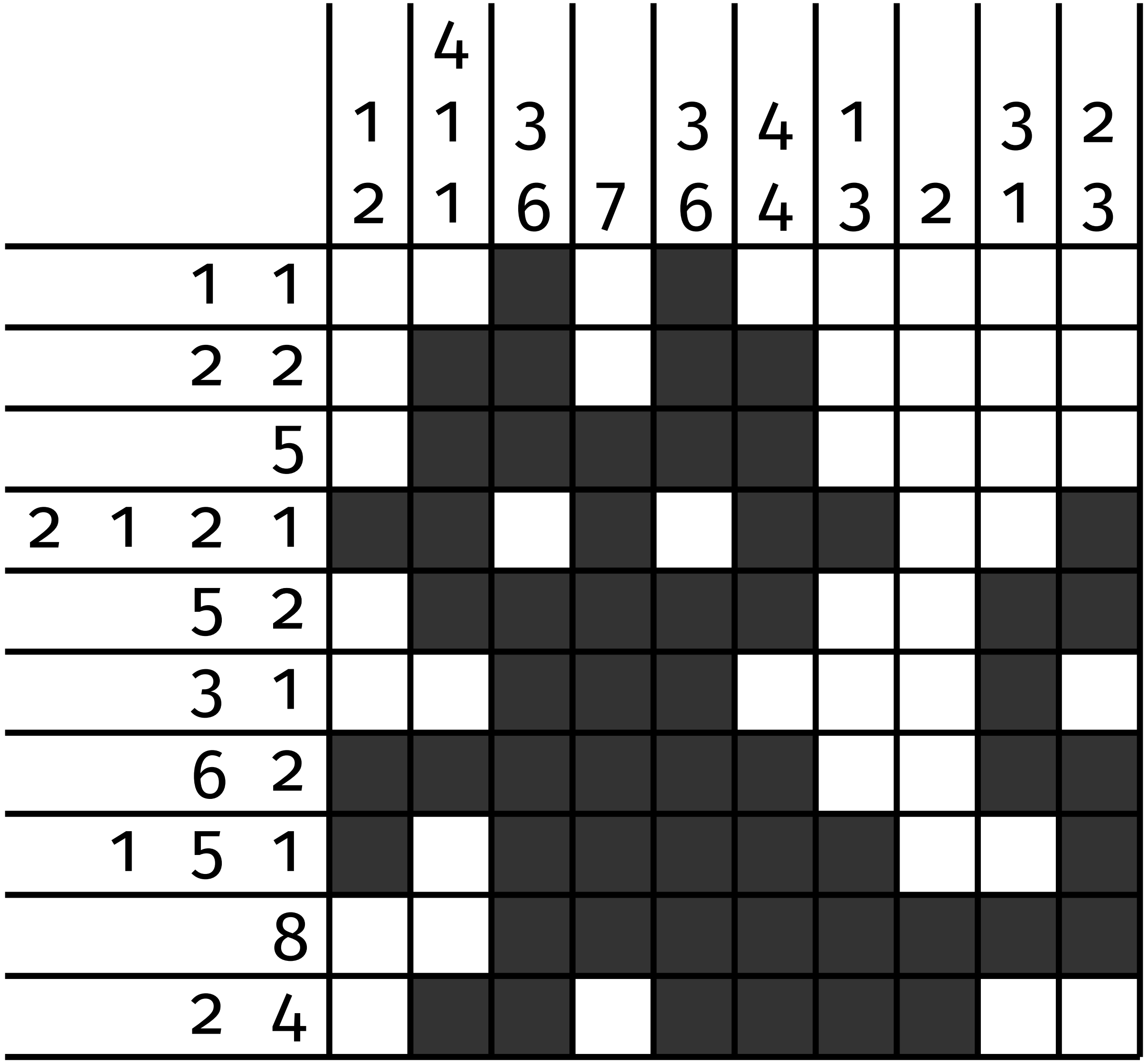
Games

Books





Entertainment



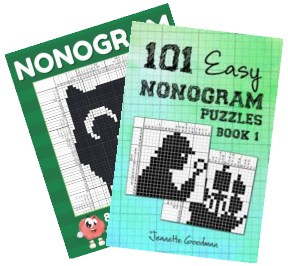
	1 2	4 1 1	3 6	7	3 6	4 4	1 3	2	3 1	2 3
1 1			■		■					
2 2		■	■		■	■				
5		■	■	■	■	■				
2 1 2 1	■	■		■		■	■			■
5 2		■	■	■	■	■			■	■
3 1			■	■	■				■	
6 2	■	■	■	■	■	■			■	■
1 5 1	■		■	■	■	■	■			■
8			■	■	■	■	■	■	■	■
2 4		■	■		■	■	■	■		

Nonograms (Griddlers, 判じ絵, Picross,...)

Popular with consumers...

Games

Books



Entertainment

...and scientists

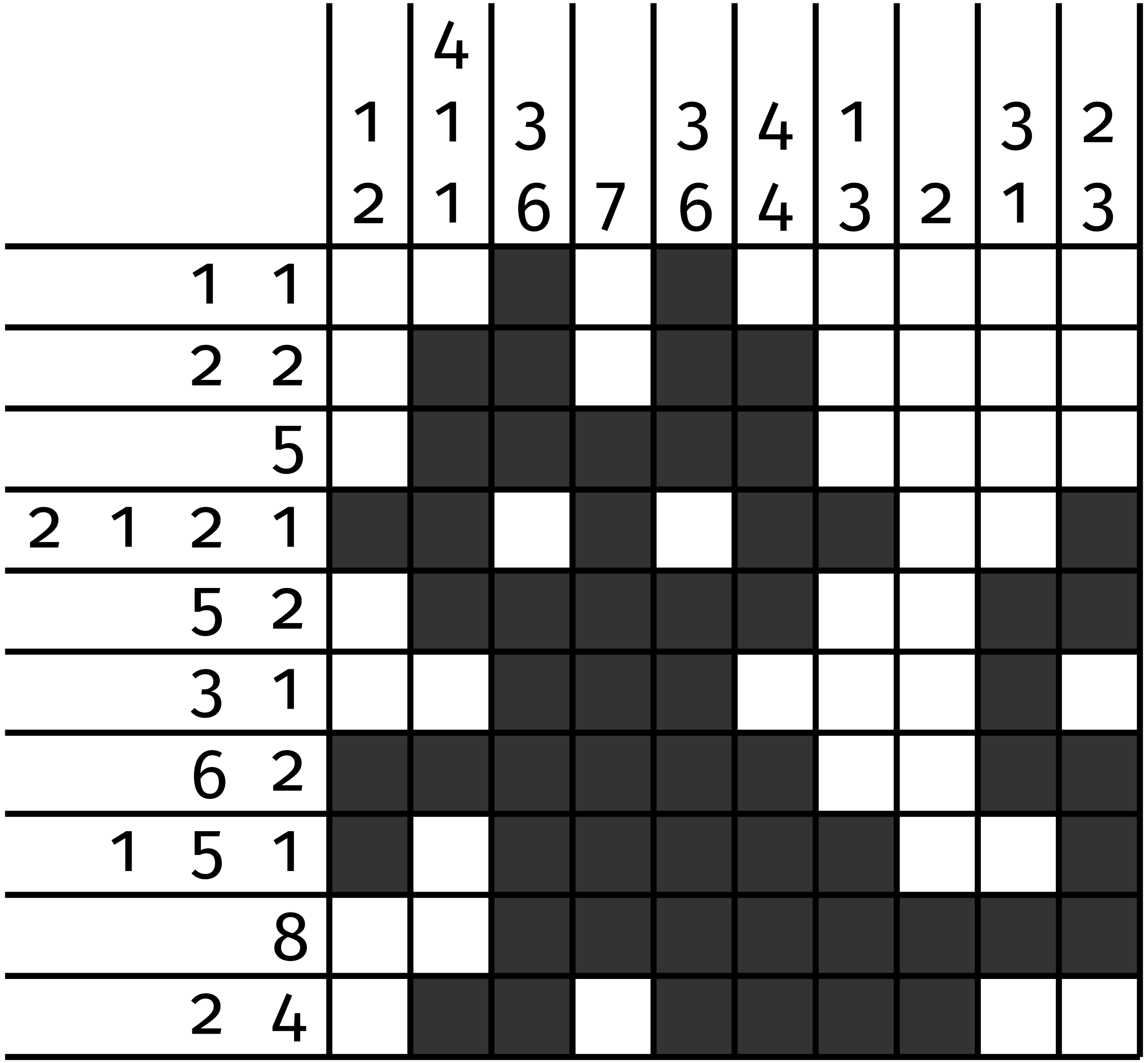
[Batenburg & Kosters, '09]

[Yu et al., '11]

[Batenburg & Kosters, '12]

[Berend et al., '14]

[Chen & Lin, '19]

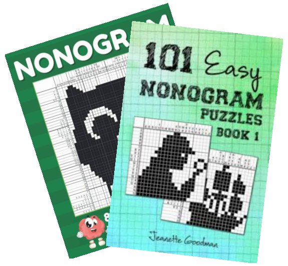


	1 2	4 1 1	3 6	7	3 6	4 4	1 3	2	3 1	2 3
1 1			■		■					
2 2		■	■		■	■				
5		■	■	■	■	■				
2 1 2 1	■	■		■		■	■			■
5 2		■	■	■	■	■			■	■
3 1			■	■	■				■	
6 2	■	■	■	■	■	■			■	■
1 5 1	■		■	■	■	■	■			■
8			■	■	■	■	■	■	■	■
2 4		■	■		■	■	■	■		

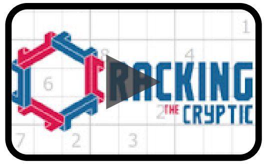
Nonograms (Griddlers, 判じ絵, Picross,...)

Popular with consumers...

Games



Books

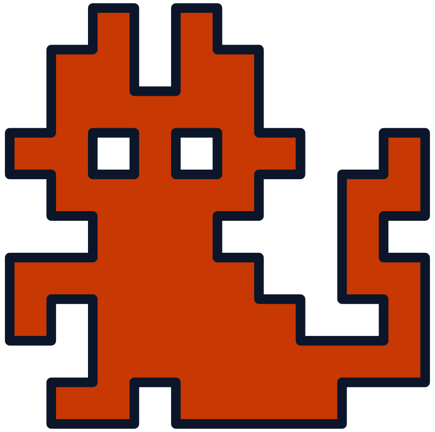


Entertainment

...and scientists

[Batenburg & Kosters, ’09]
[Yu et al., ’11]
[Batenburg & Kosters, ’12]
[Berend et al., ’14]
[Chen & Lin, ’19]

Pixel image

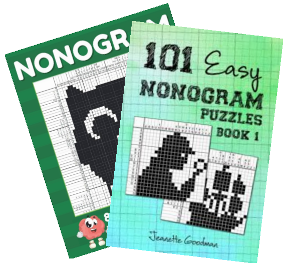


Nonograms (Griddlers, 判じ絵, Picross,...)

Popular with consumers...

Games

Books



Entertainment

...and scientists

[Batenburg & Kosters, ’09]

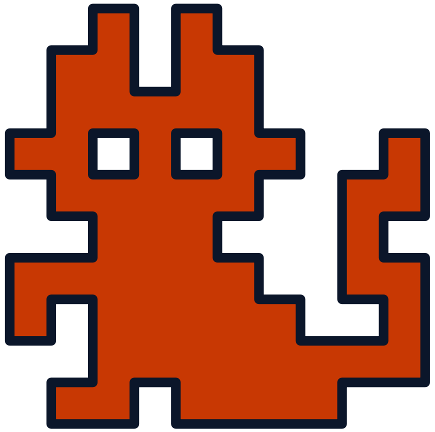
[Yu et al., ’11]

[Batenburg & Kosters, ’12]

[Berend et al., ’14]

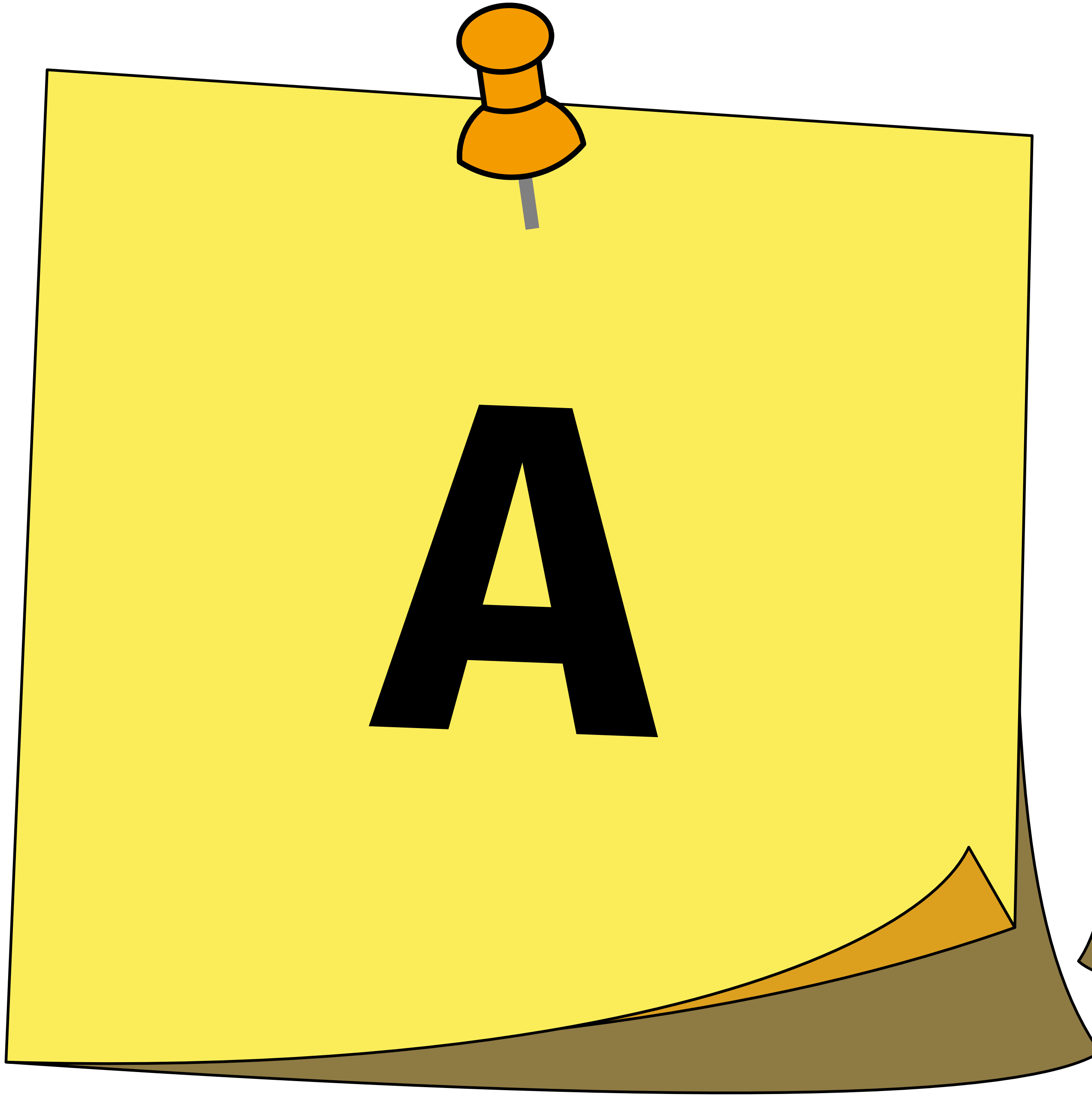
[Chen & Lin, ’19]

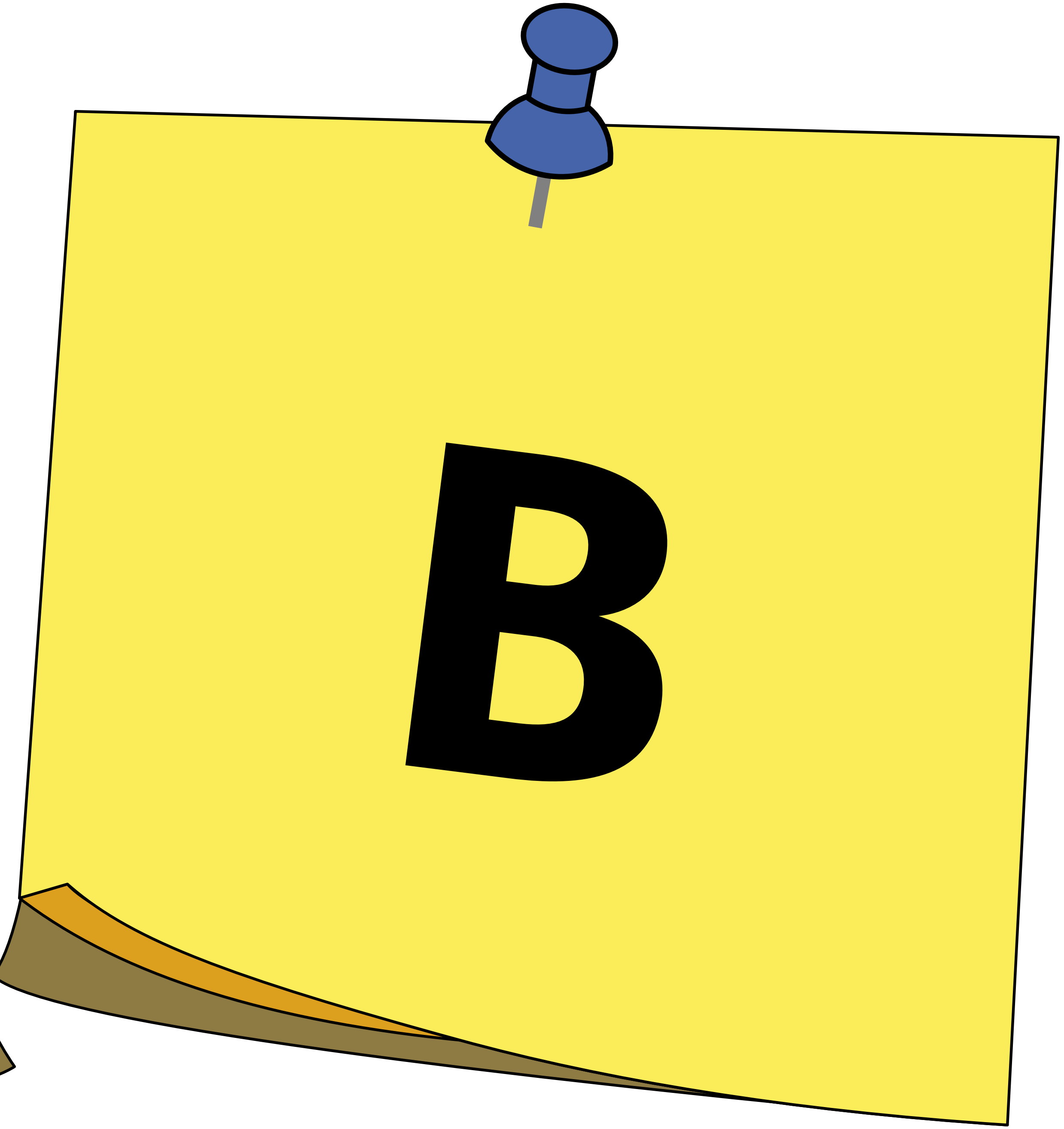
Pixel image



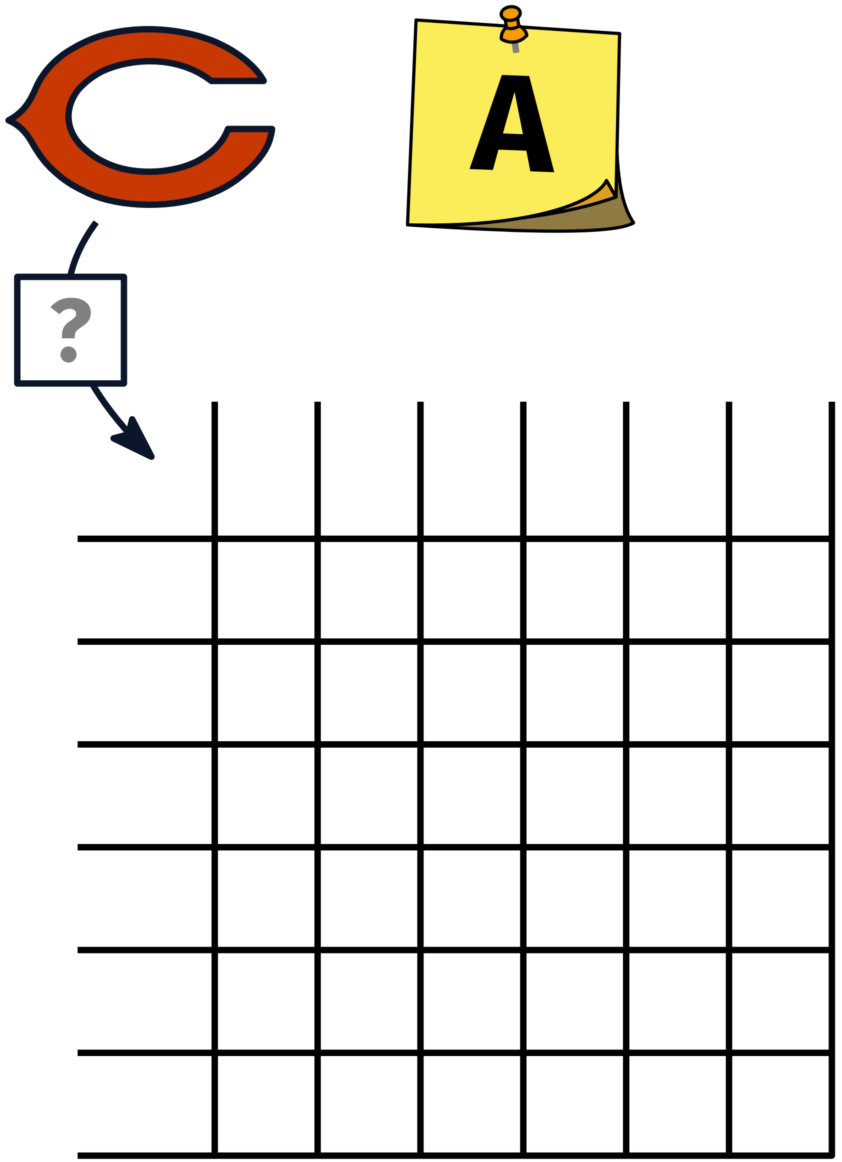
Curved image?

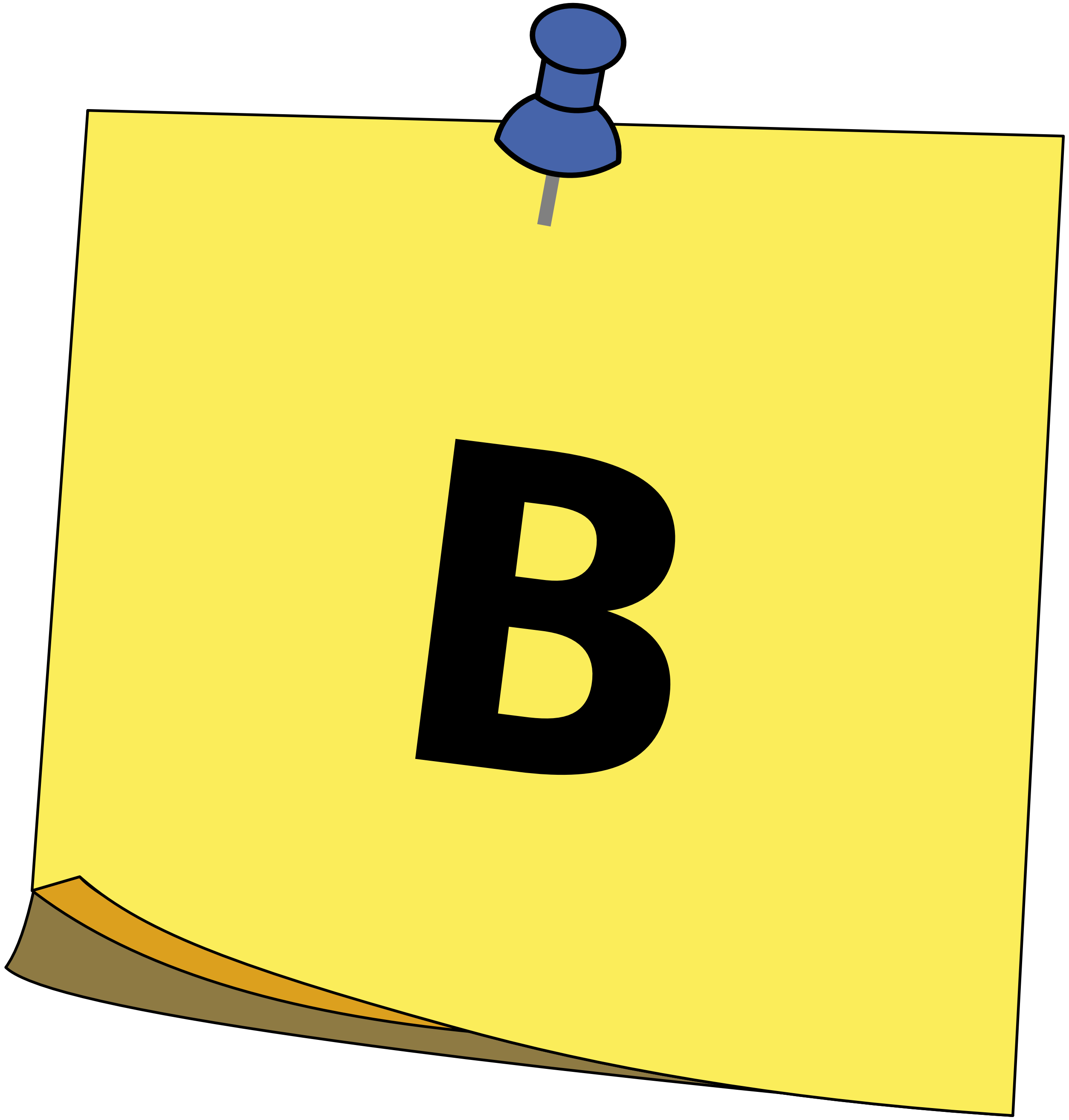
Nonograms (Griddlers, 判じ絵, Picross,...)



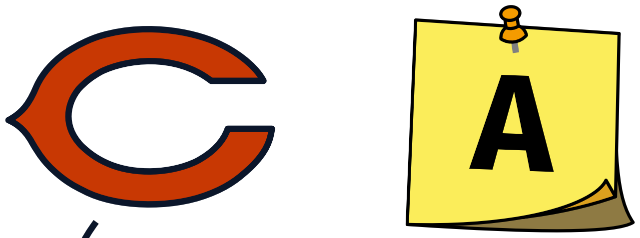


Nonograms (Griddlers, 判じ絵, Picross,...)

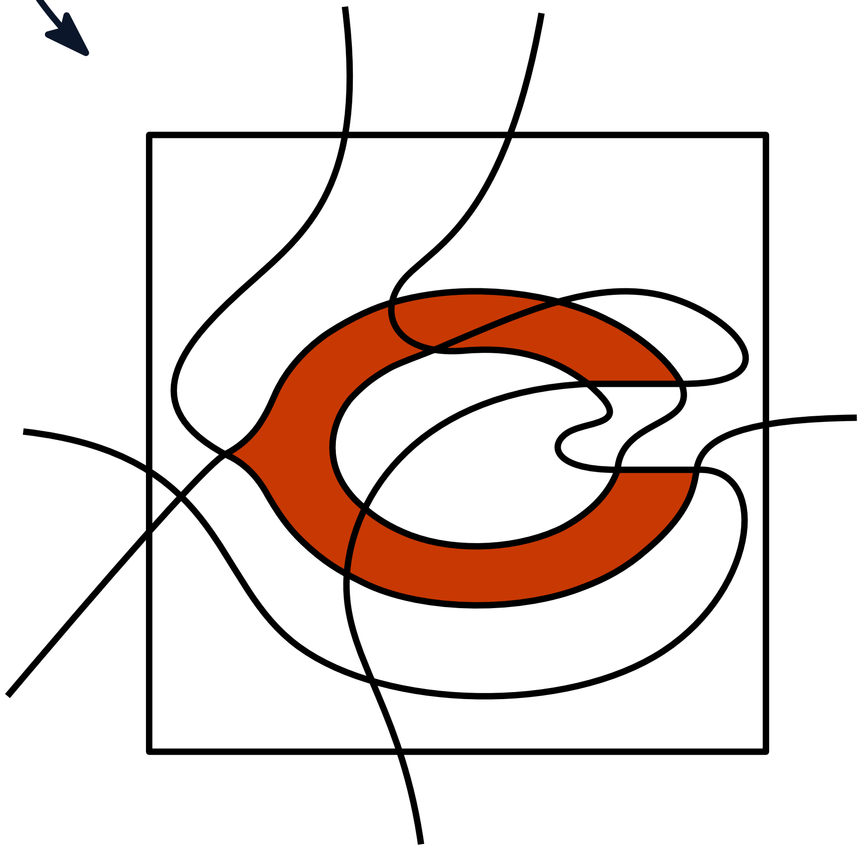


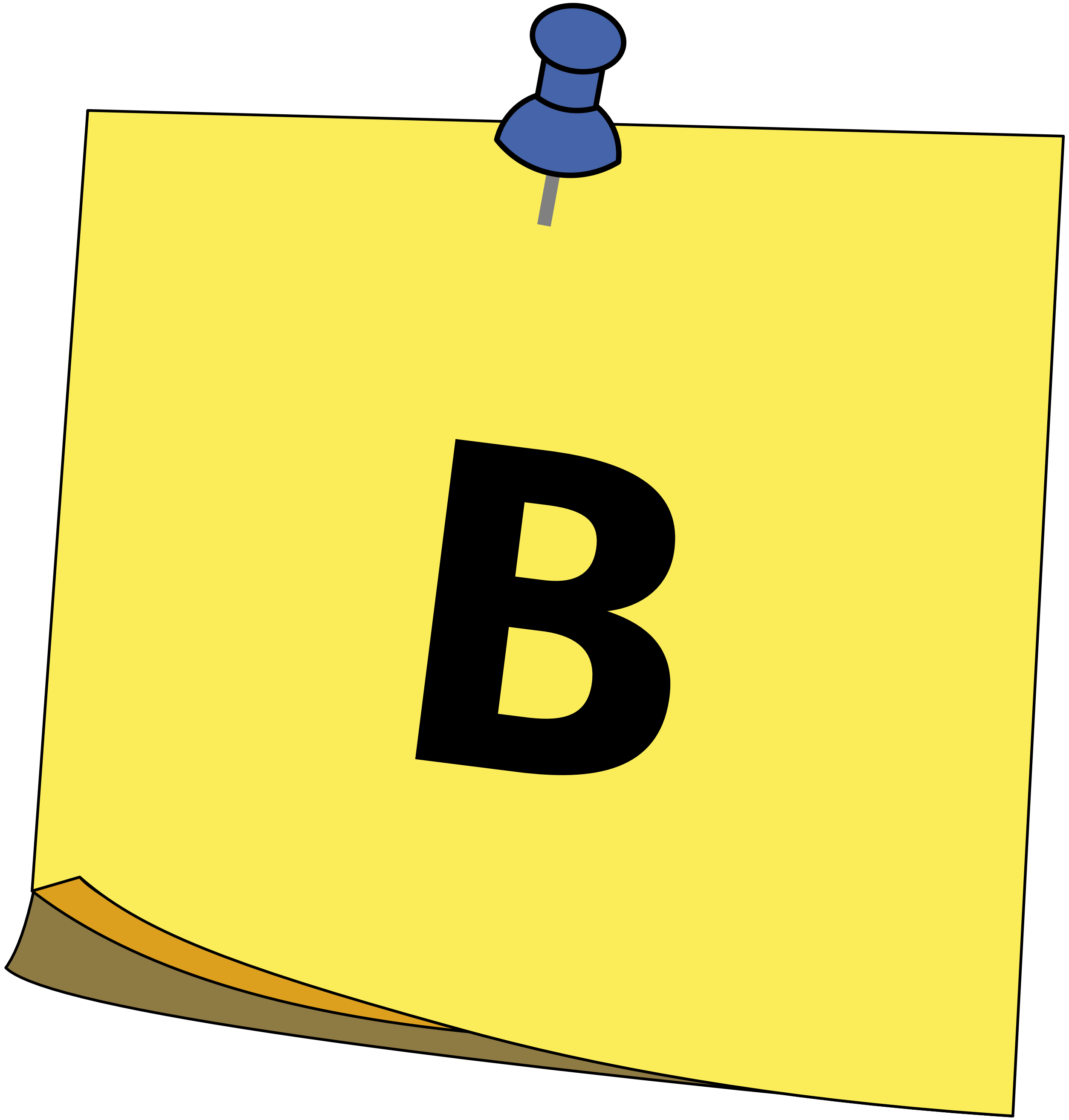


Nonograms (Griddlers, 判じ絵, Picross,...)

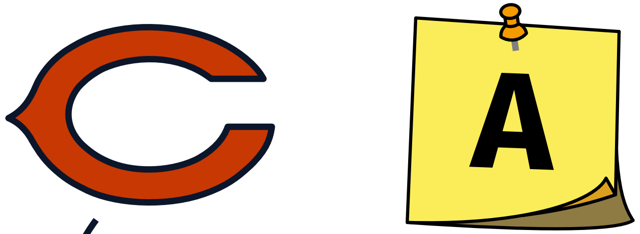


☑ Use a **curve arrangement**

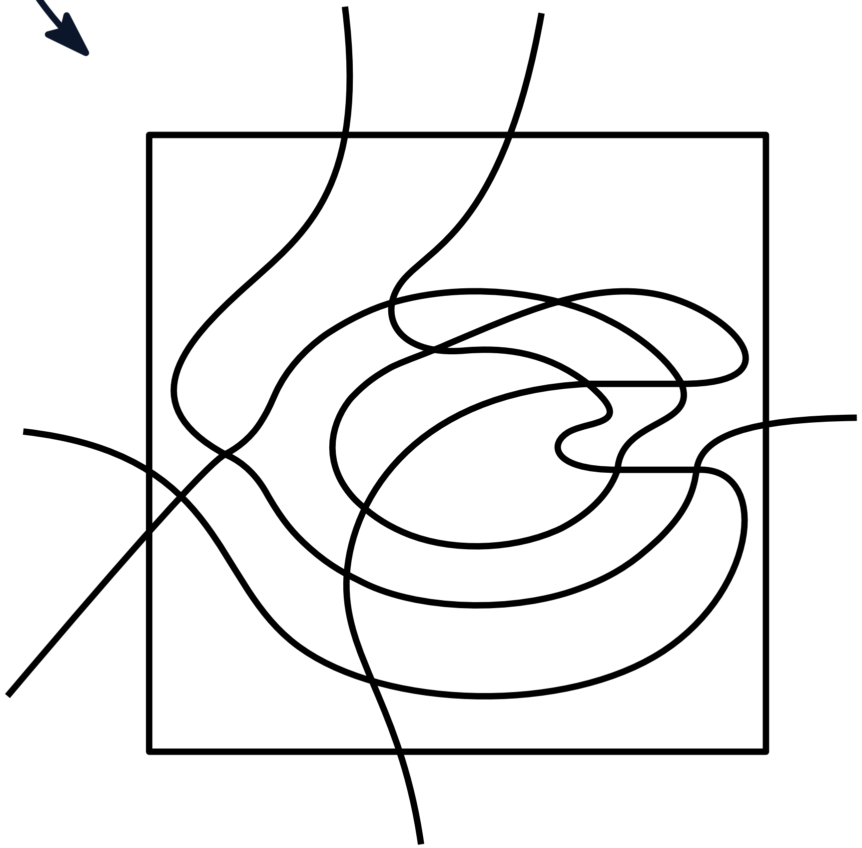


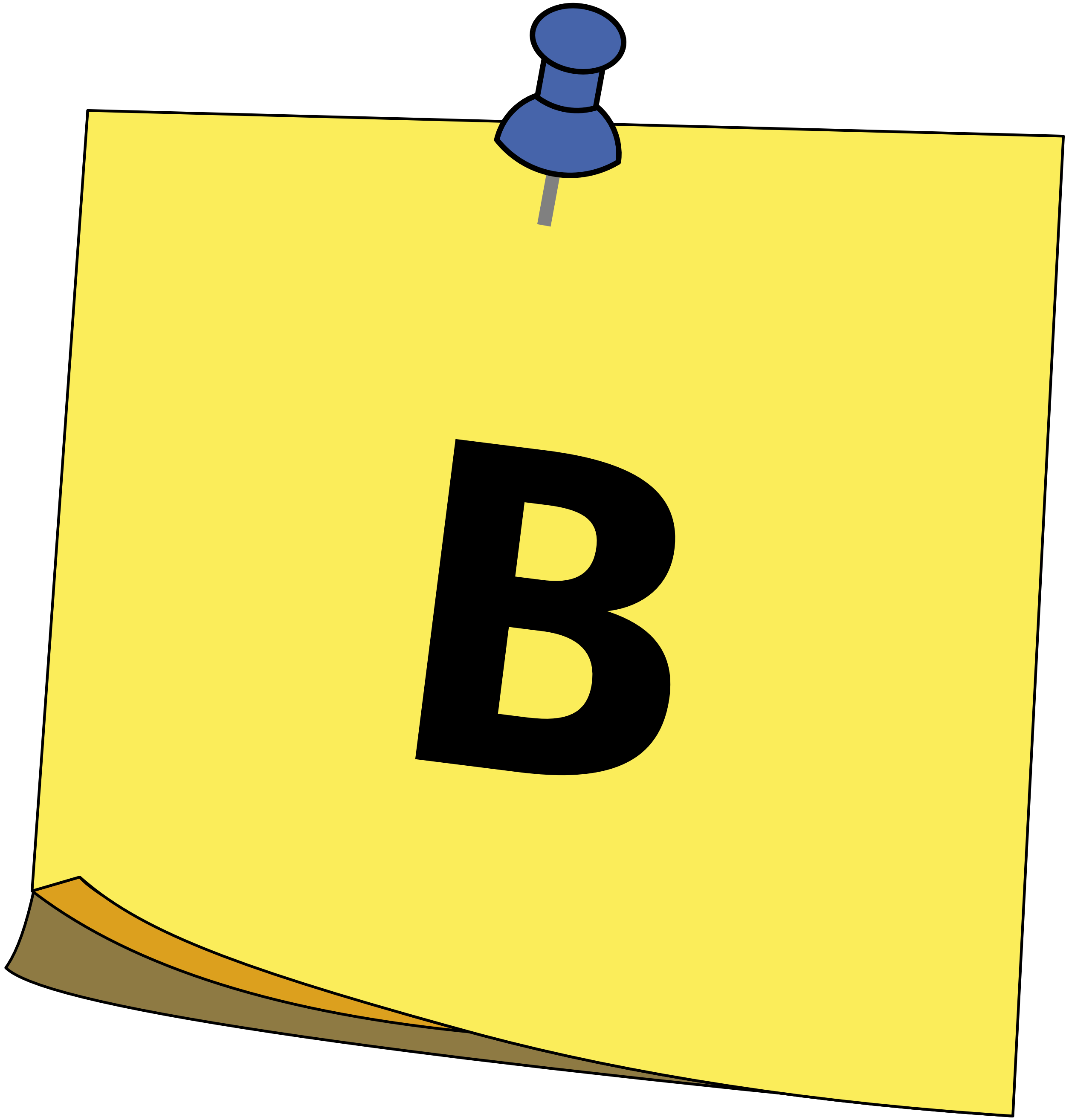


Nonograms (Griddlers, 判じ絵, Picross,...)



☑ Use a **curve arrangement**

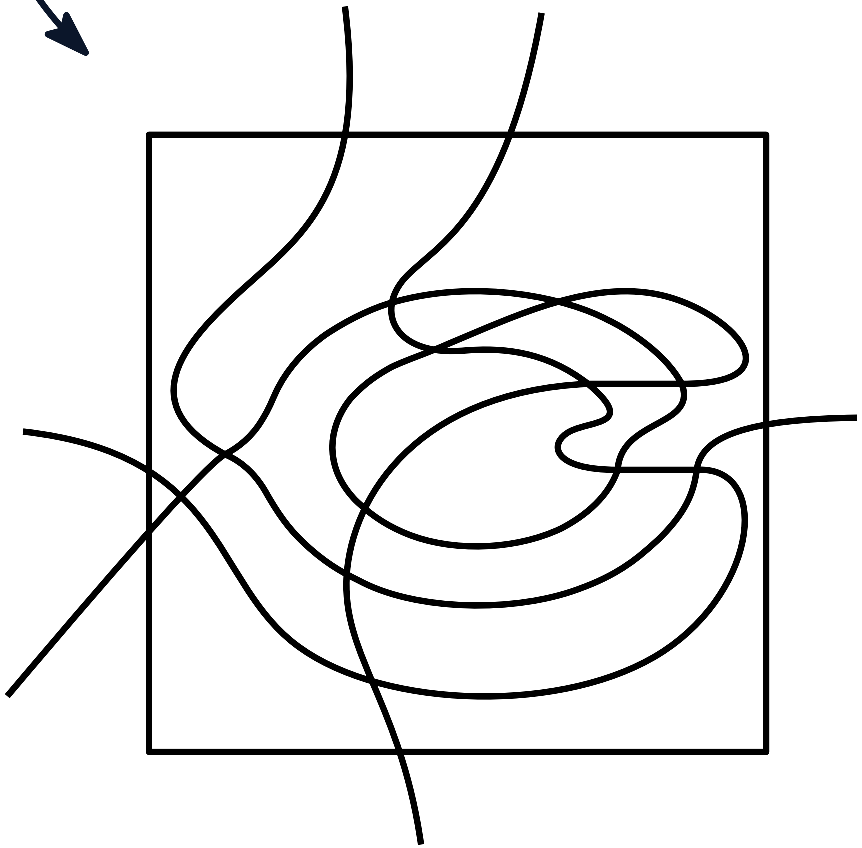




Nonograms (Griddlers, 判じ絵, Picross,...)

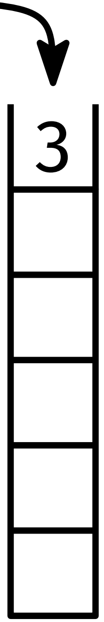
A

☑ Use a **curve arrangement**

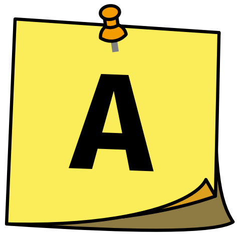


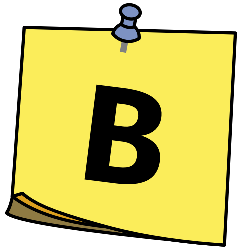
B

Three in
this column

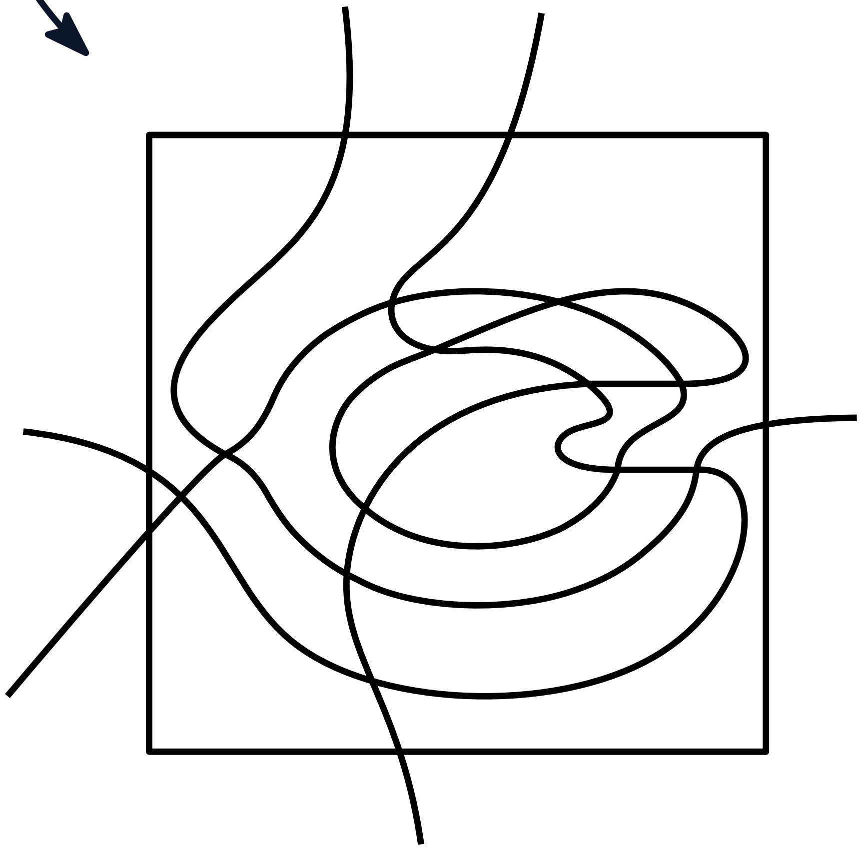


Nonograms (Griddlers, 判じ絵, Picross,...)

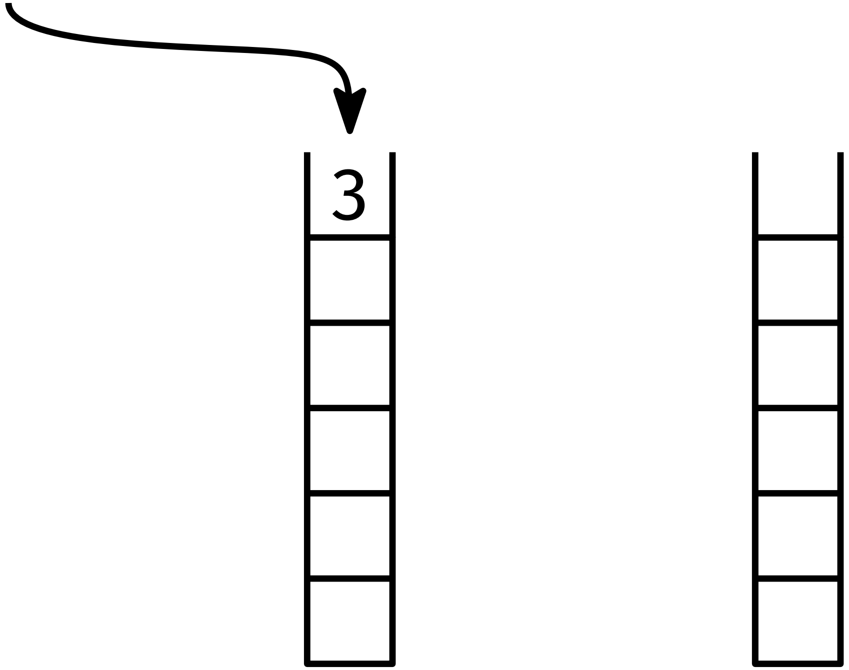




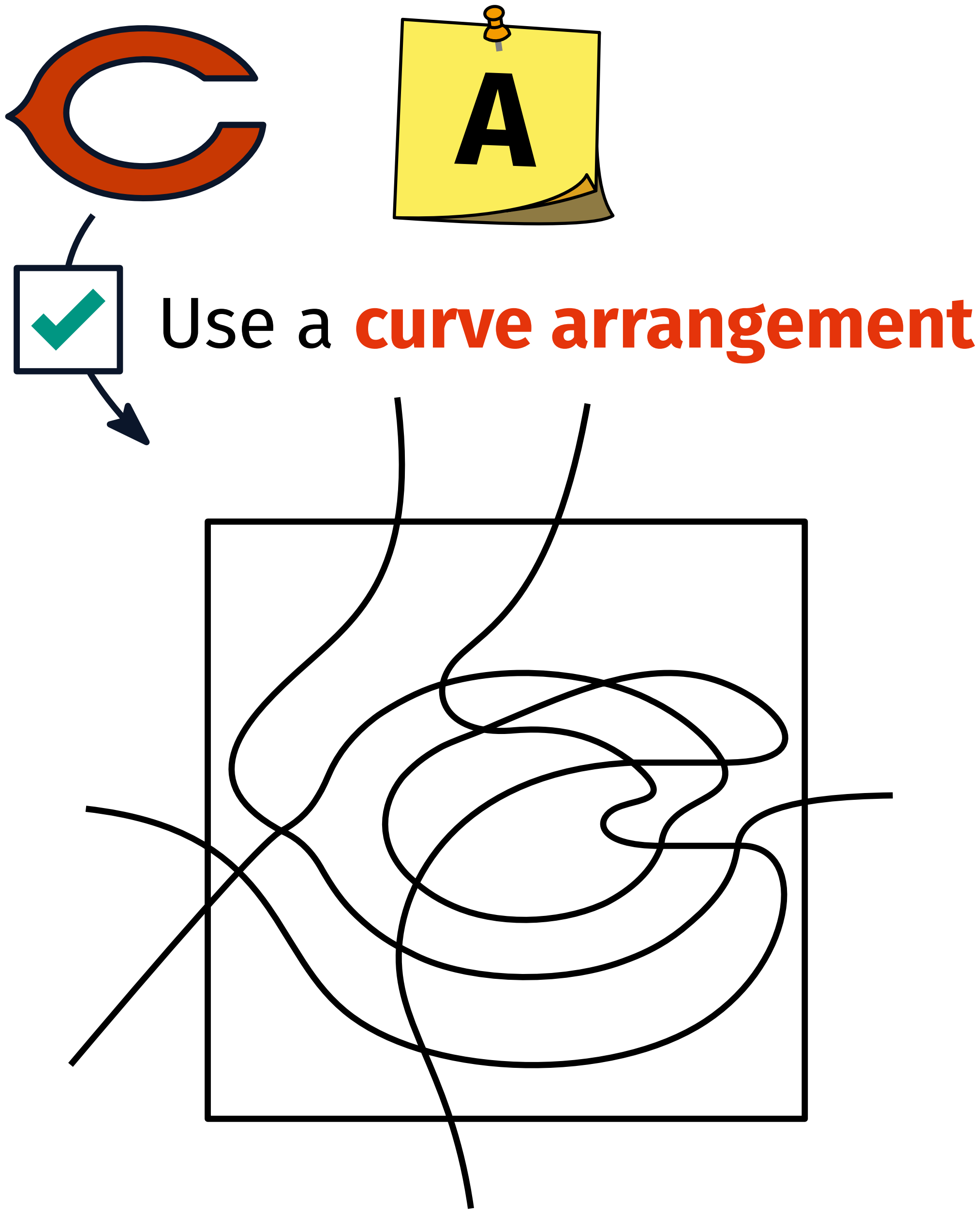
☑ Use a **curve arrangement**



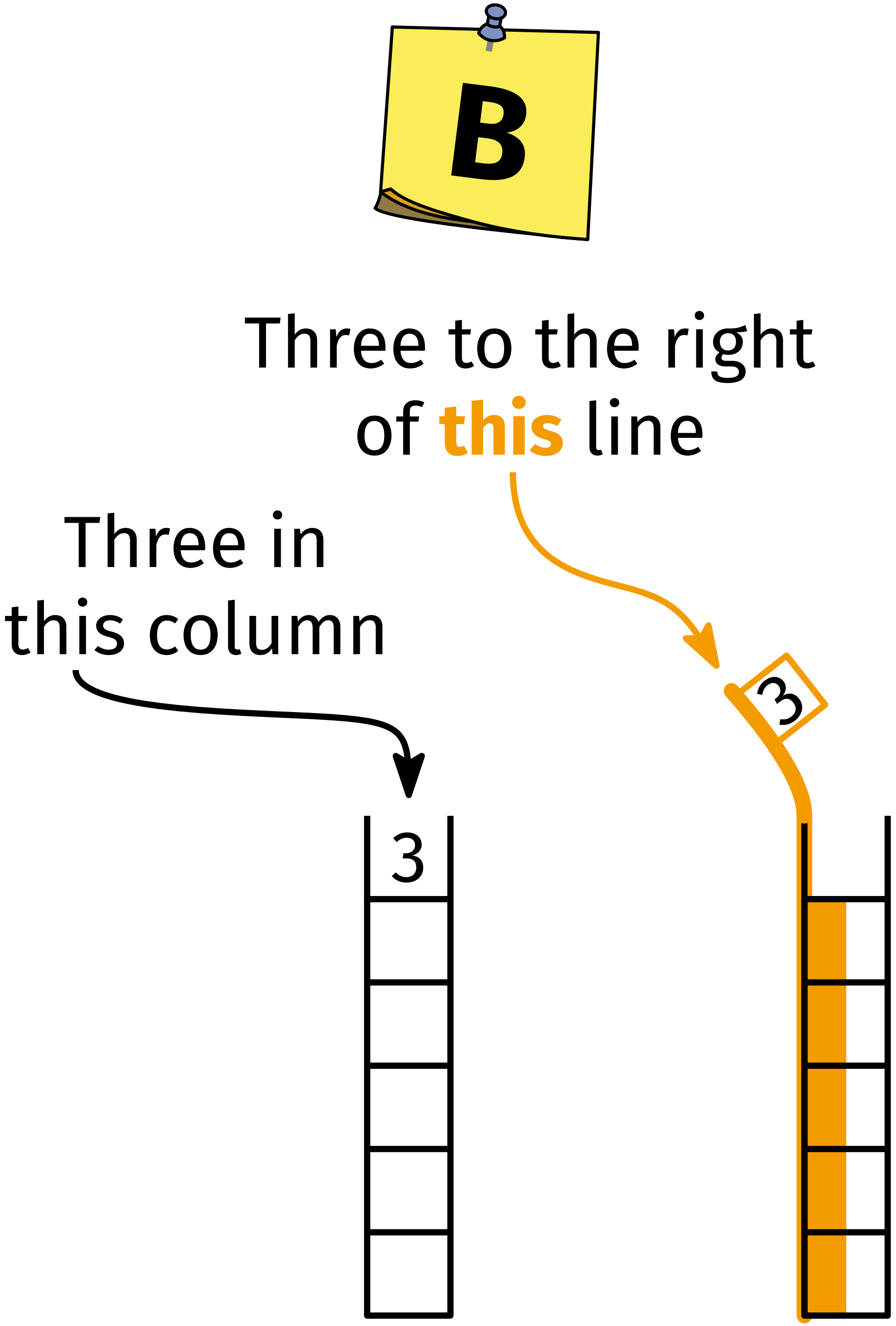
Three in
this column



Nonograms (Griddlers, 判じ絵, Picross,...)



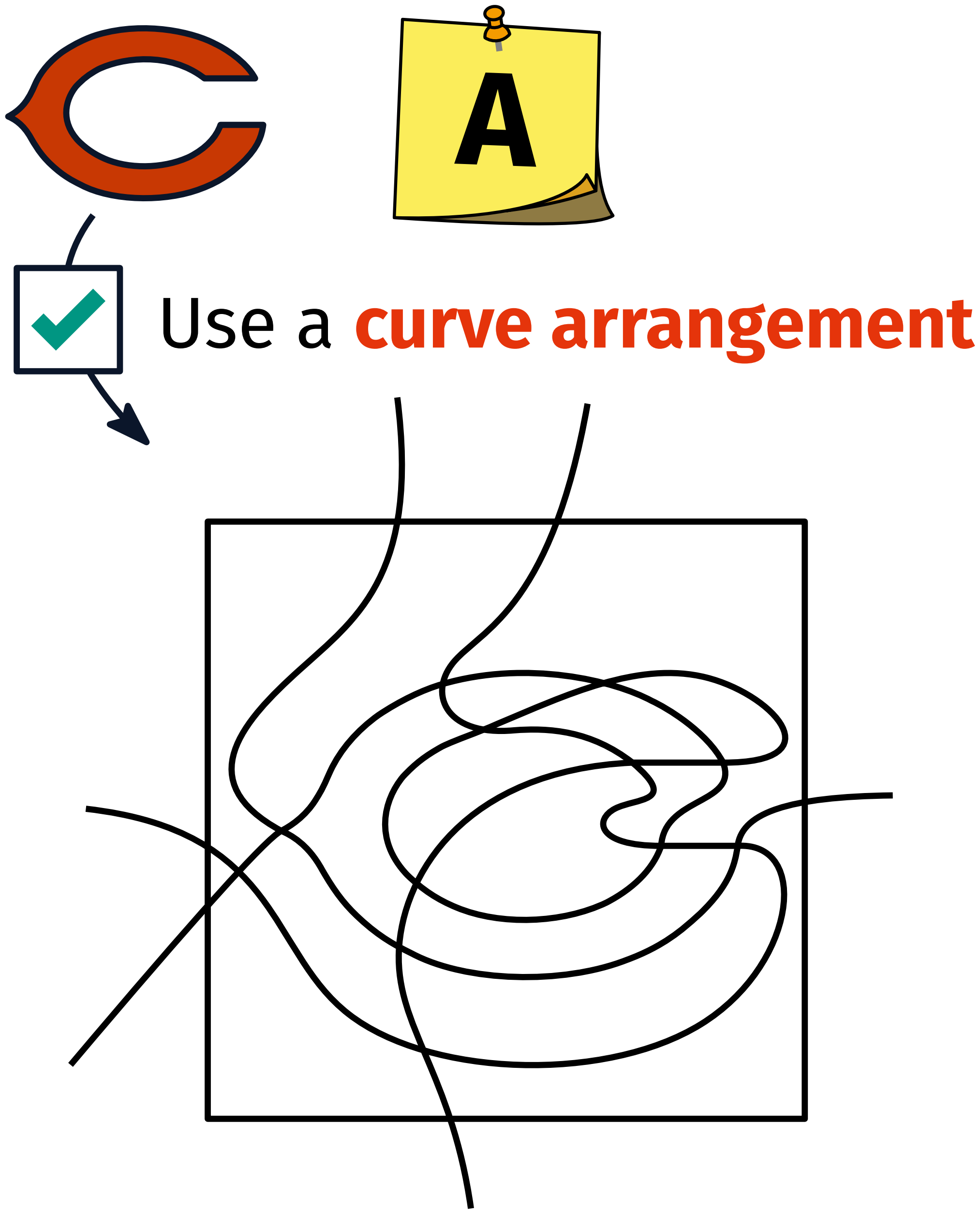
☑ Use a **curve arrangement**



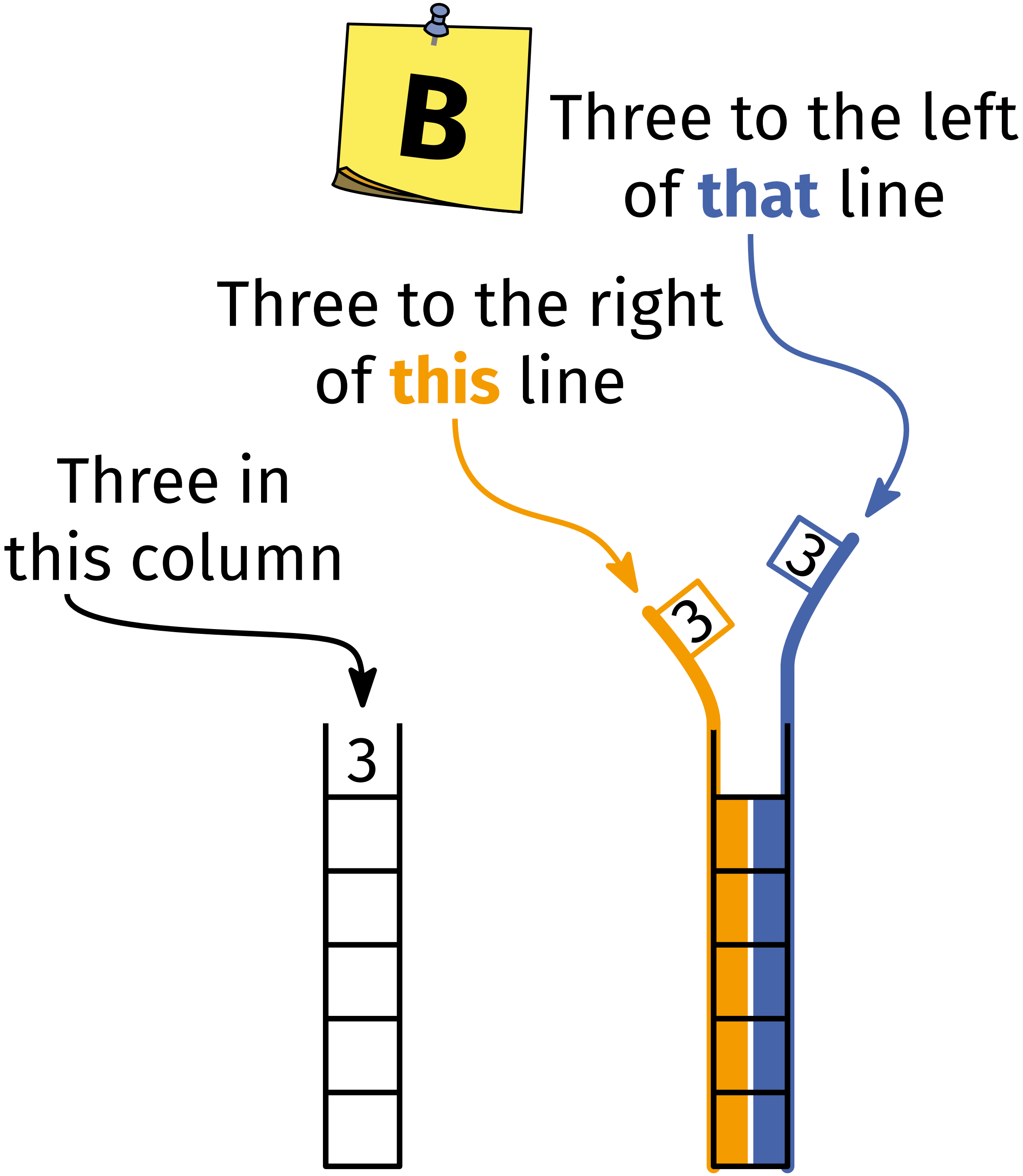
Three to the right of **this** line

Three in this column

Nonograms (Griddlers, 判じ絵, Picross,...)



☑ Use a **curve arrangement**

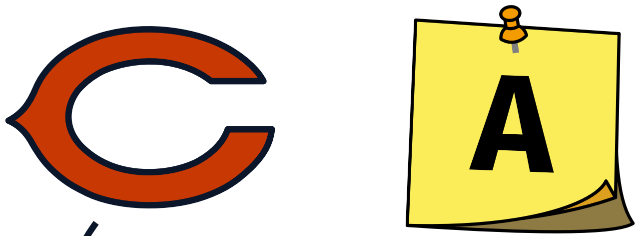


Three to the left of **that** line

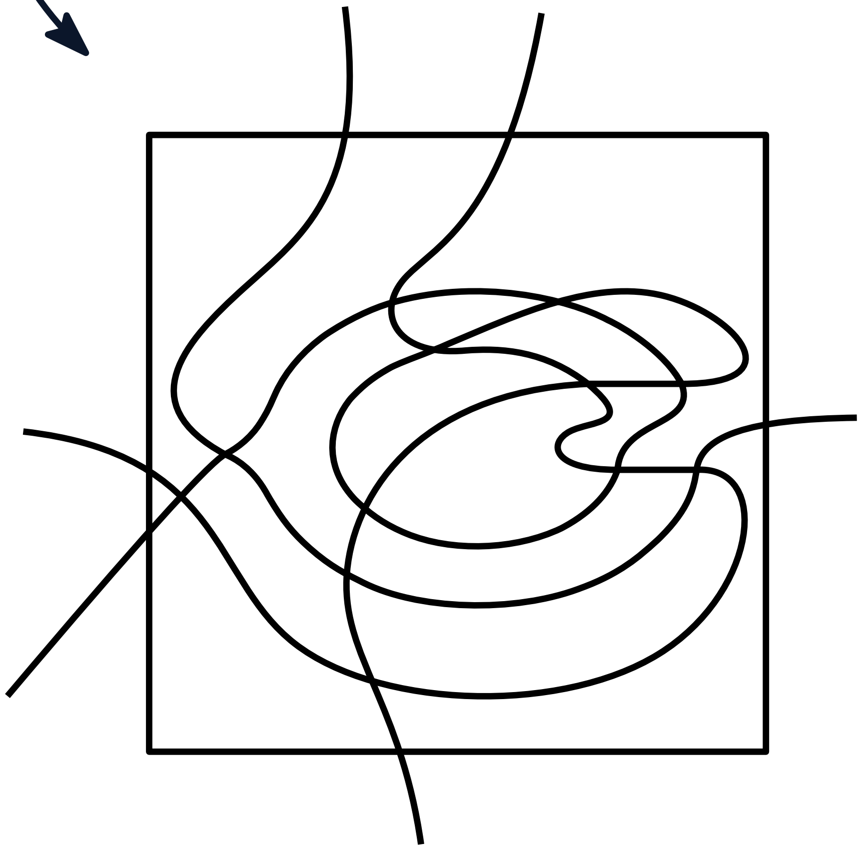
Three to the right of **this** line

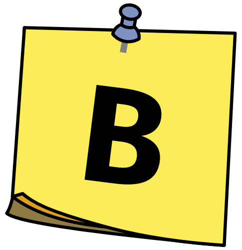
Three in this column

Nonograms (Griddlers, 判じ絵, Picross,...)



☑ Use a **curve arrangement**

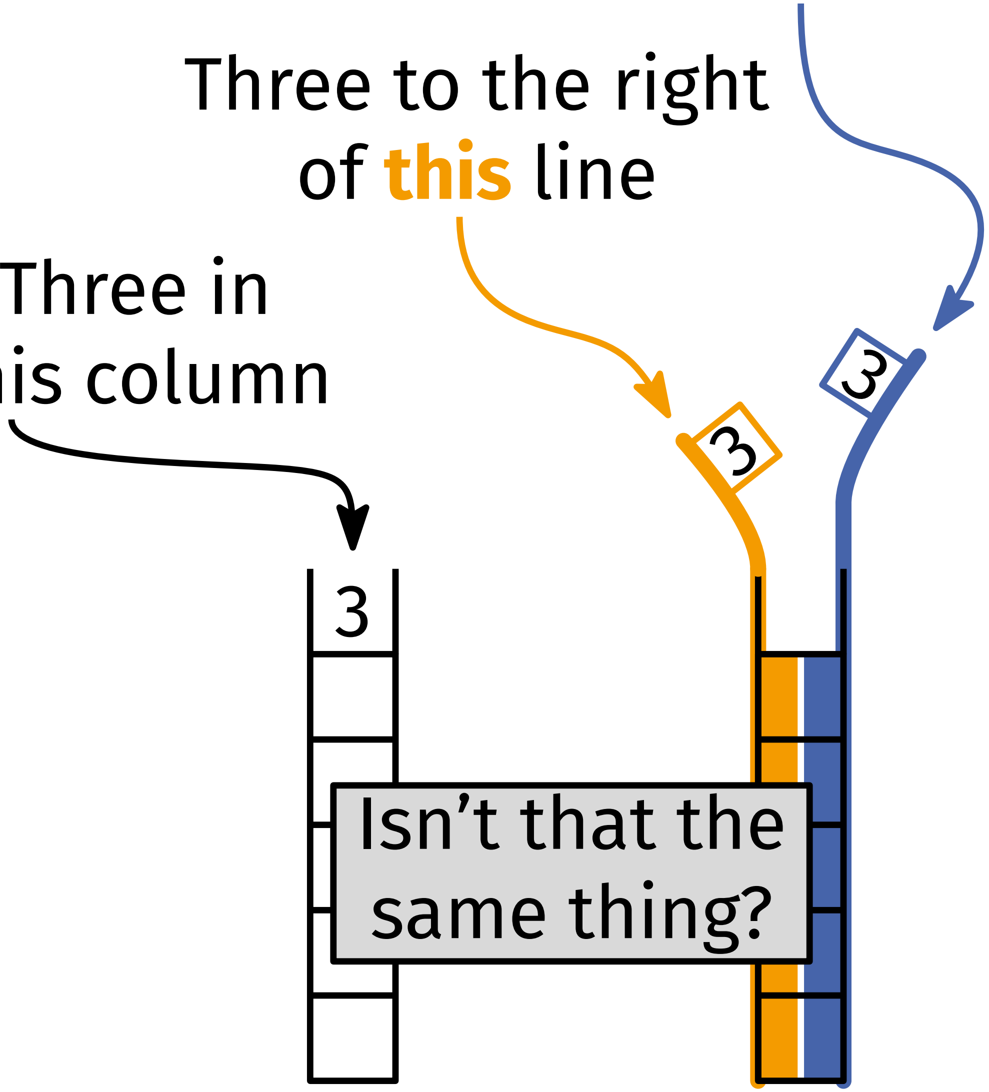




Three to the left
of **that** line

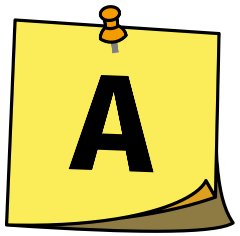
Three to the right
of **this** line

Three in
this column

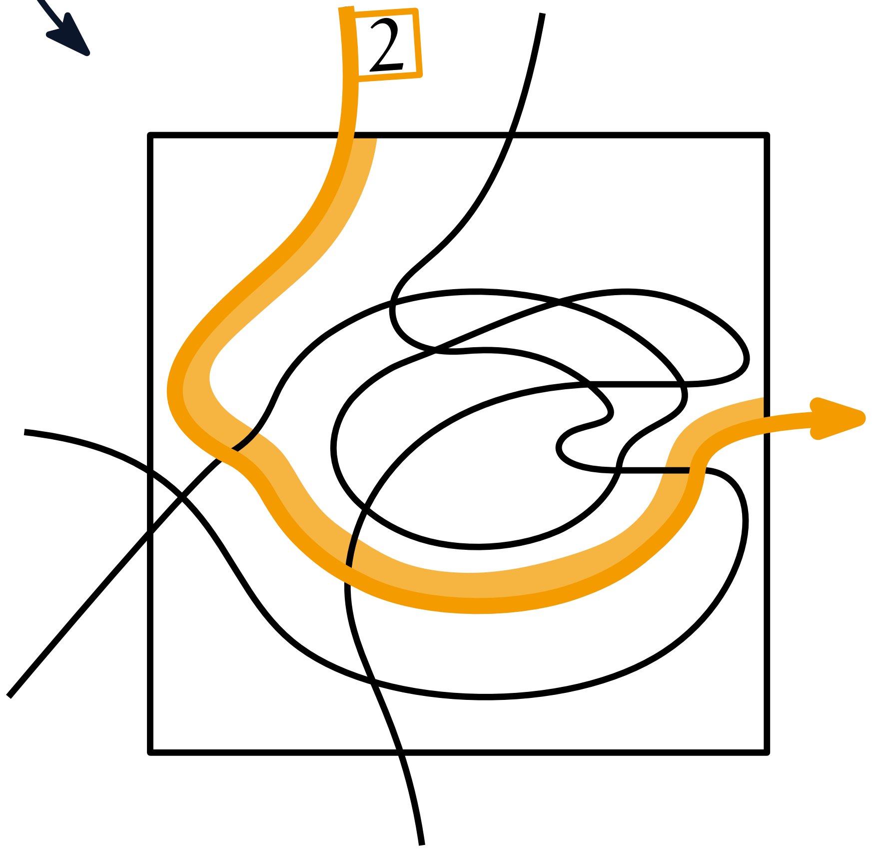


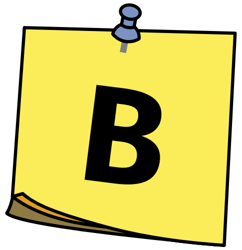
Isn’t that the
same thing?

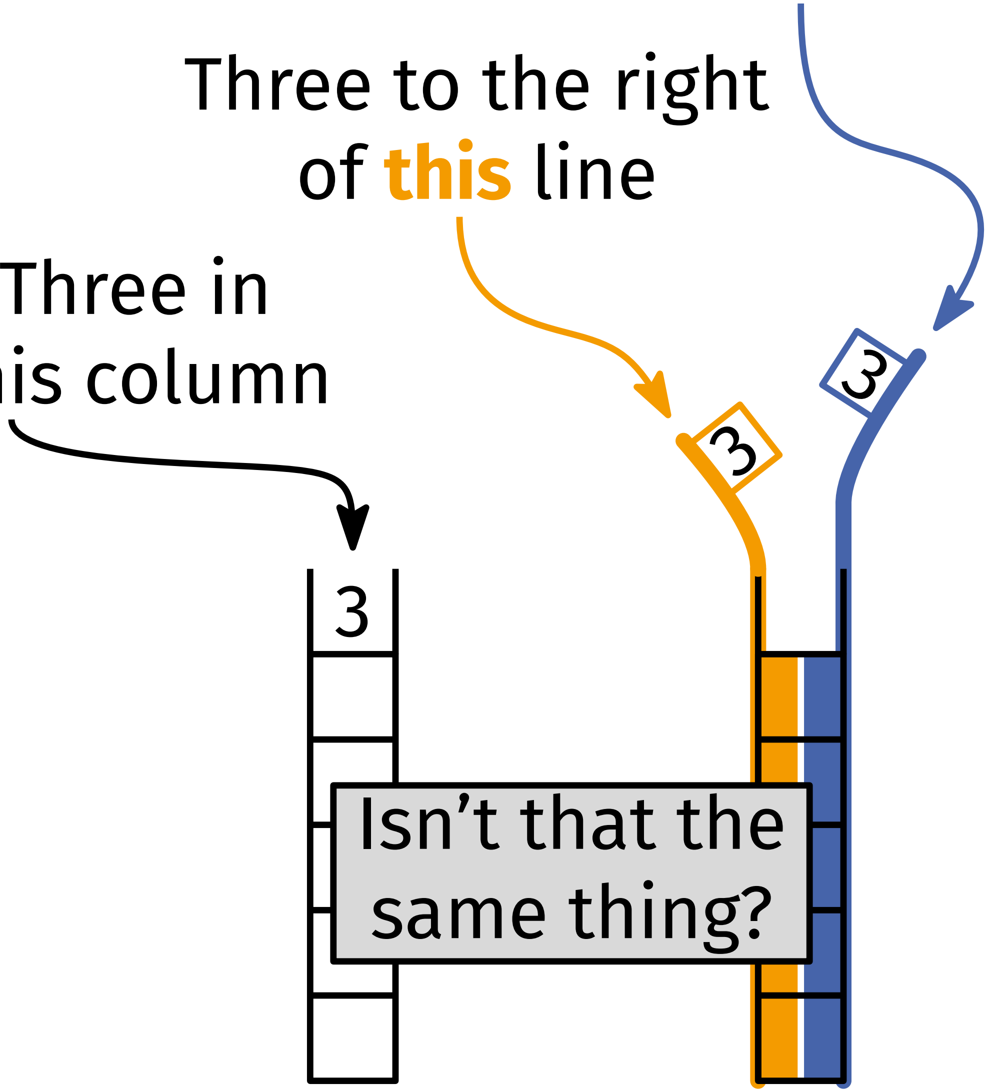
Nonograms (Griddlers, 判じ絵, Picross,...)



☑ Use a **curve arrangement**







Three to the left of **that** line

Three to the right of **this** line

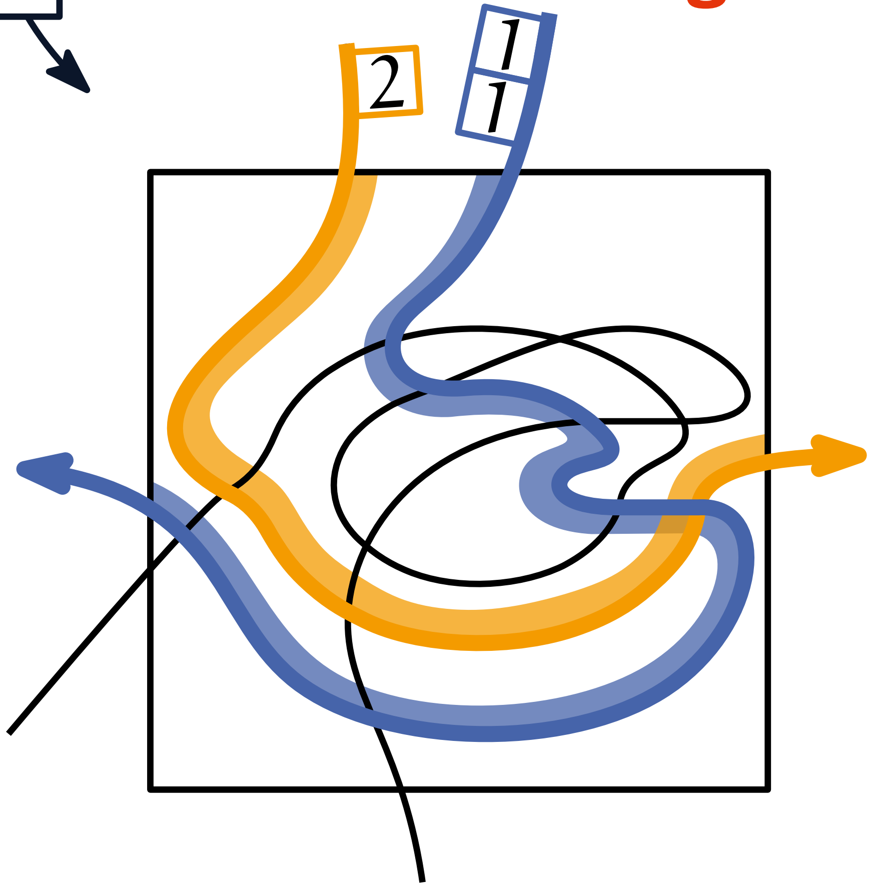
Three in this column

Isn’t that the same thing?

Nonograms (Griddlers, 判じ絵, Picross,...)

A

☑ Use a **curve arrangement**

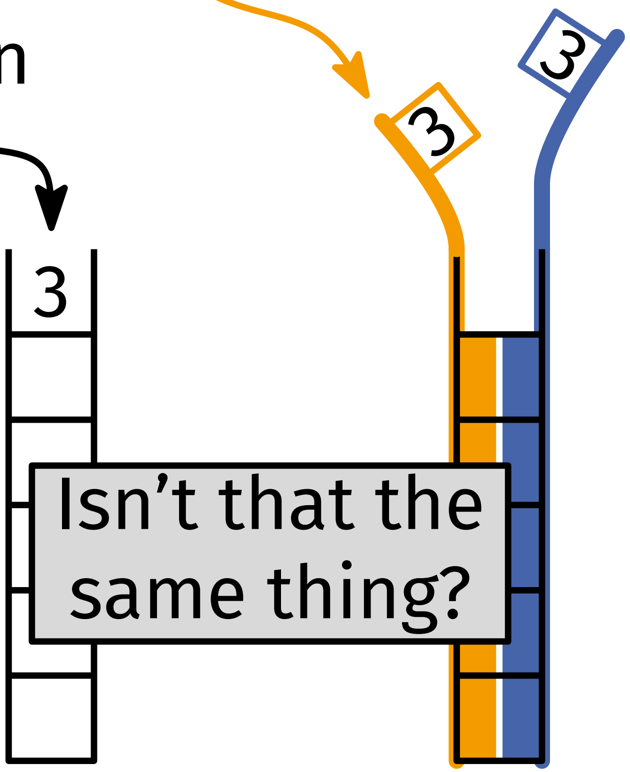


B

Three to the left
of **that** line

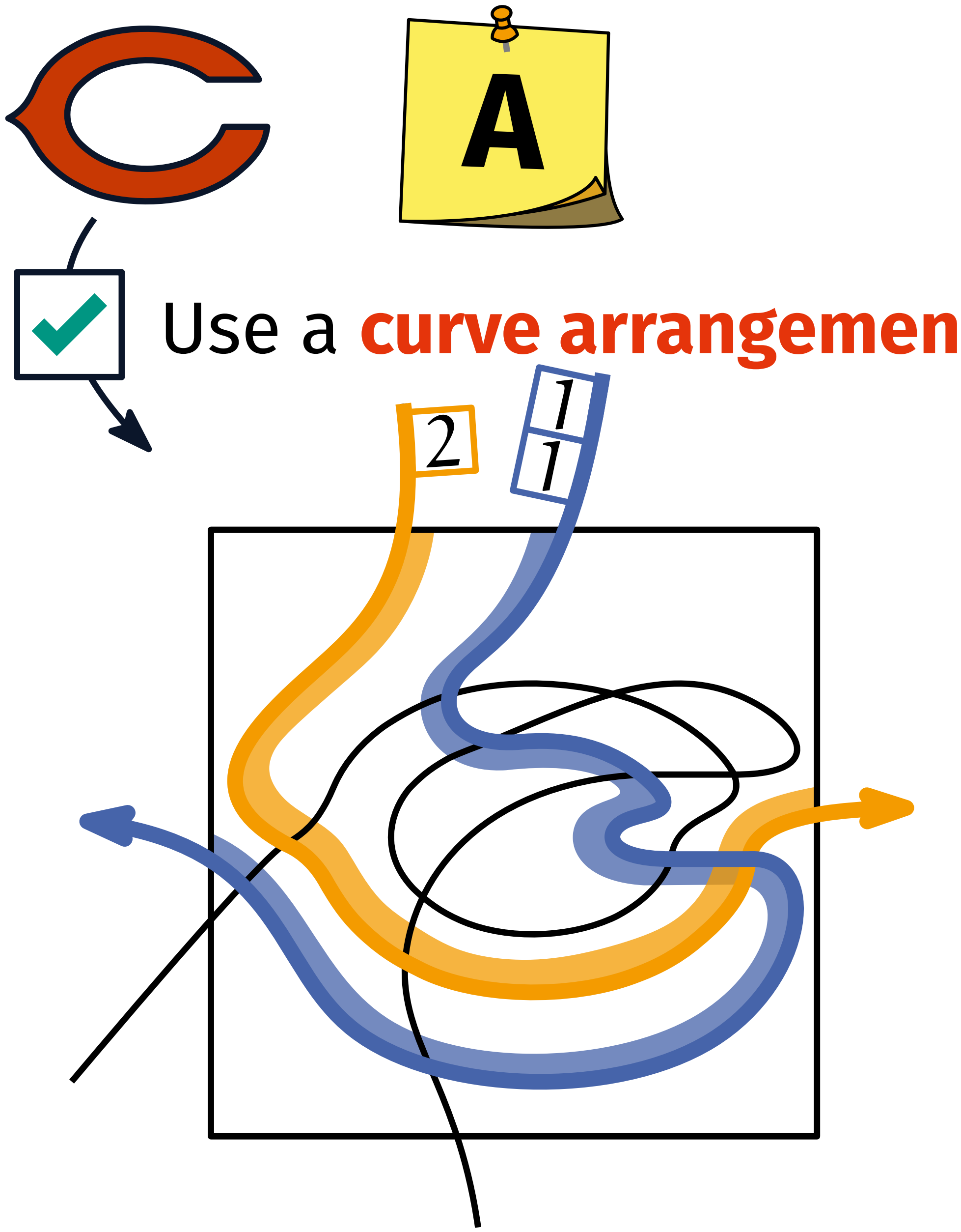
Three to the right
of **this** line

Three in
this column



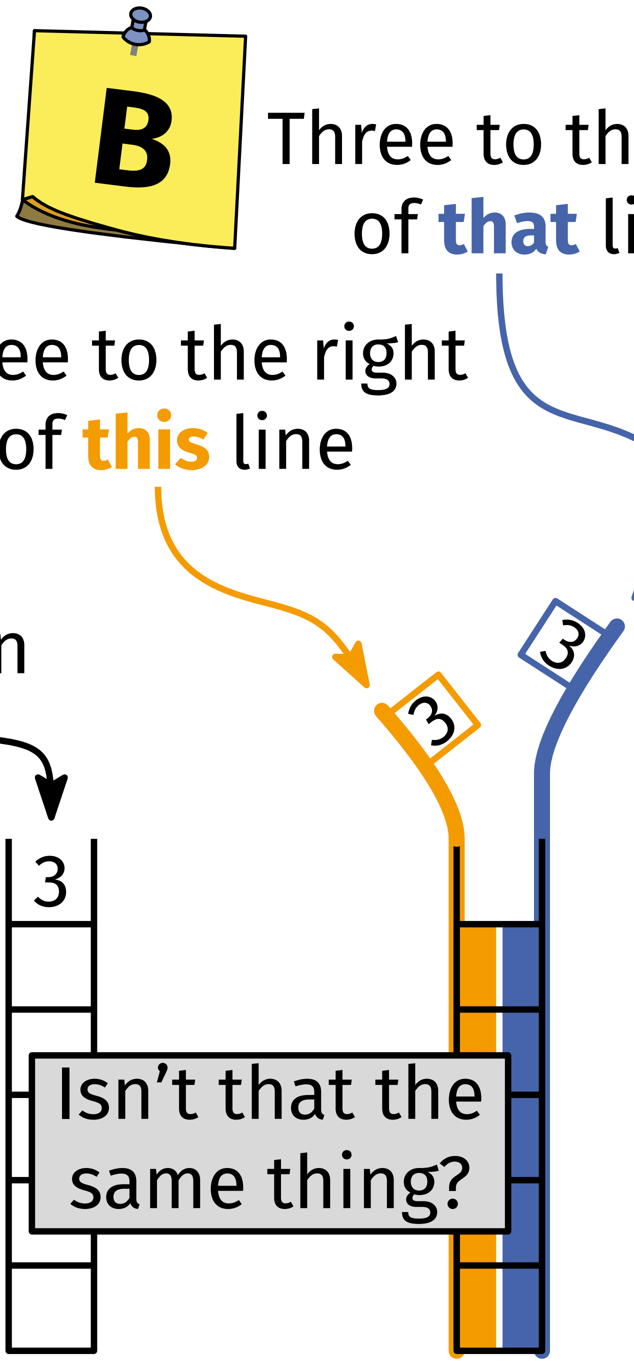
Isn't that the
same thing?

Nonograms (Griddlers, 判じ絵, Picross,...)



☑ Use a **curve arrangement**

[van de Kerkhof et al., ’19]



Three to the left of **that** line

Three to the right of **this** line

Three in this column

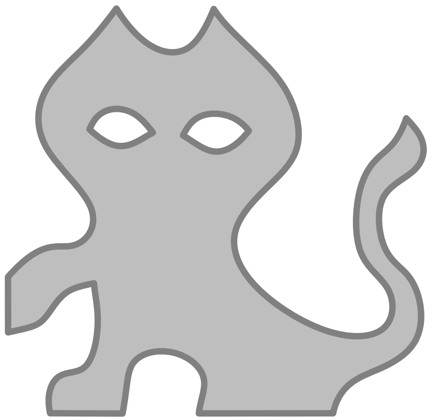
Isn’t that the same thing?

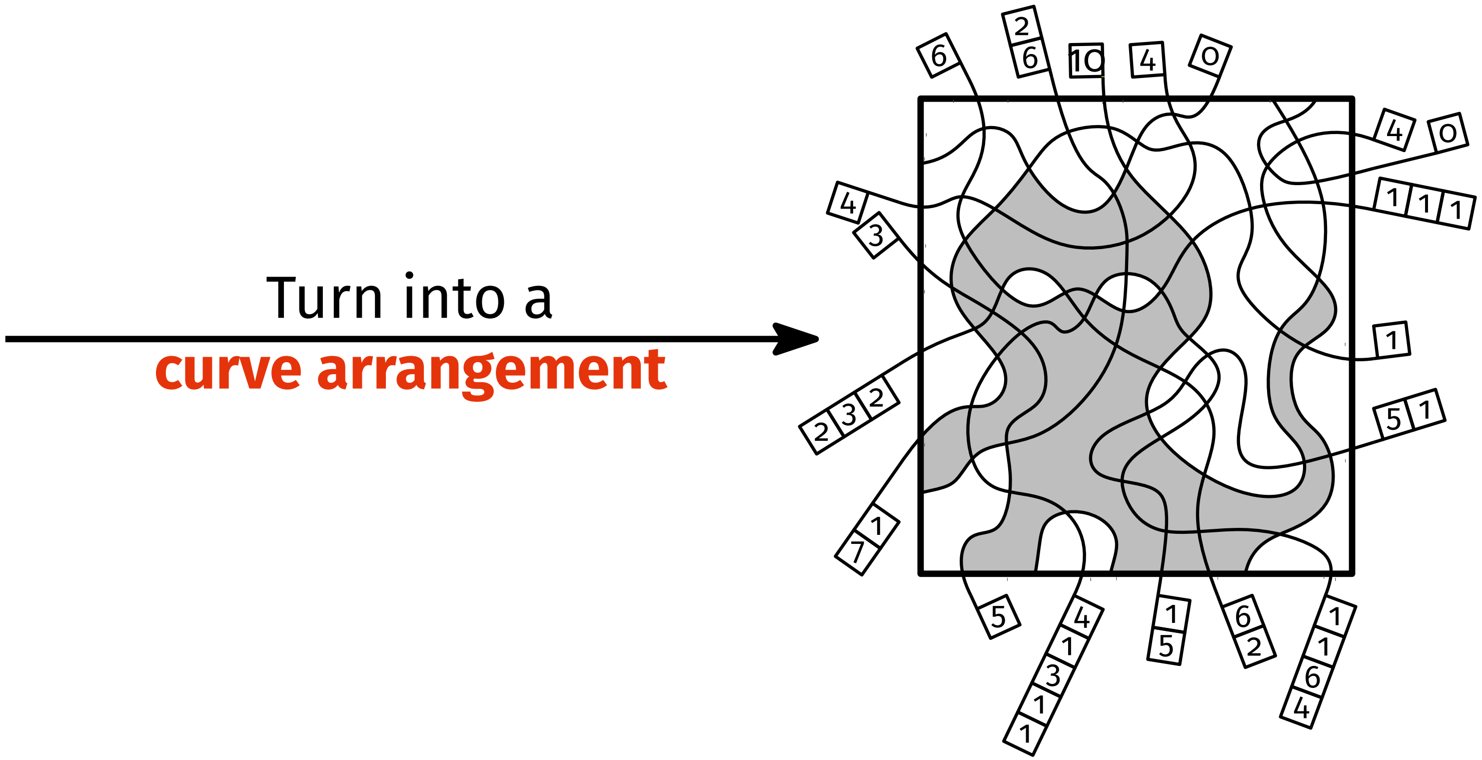
Curved Nonograms

Given a shape...

curved

...compute a curved nonogram





Turn into a **curve arrangement**

Good: Previous work for generating nonograms

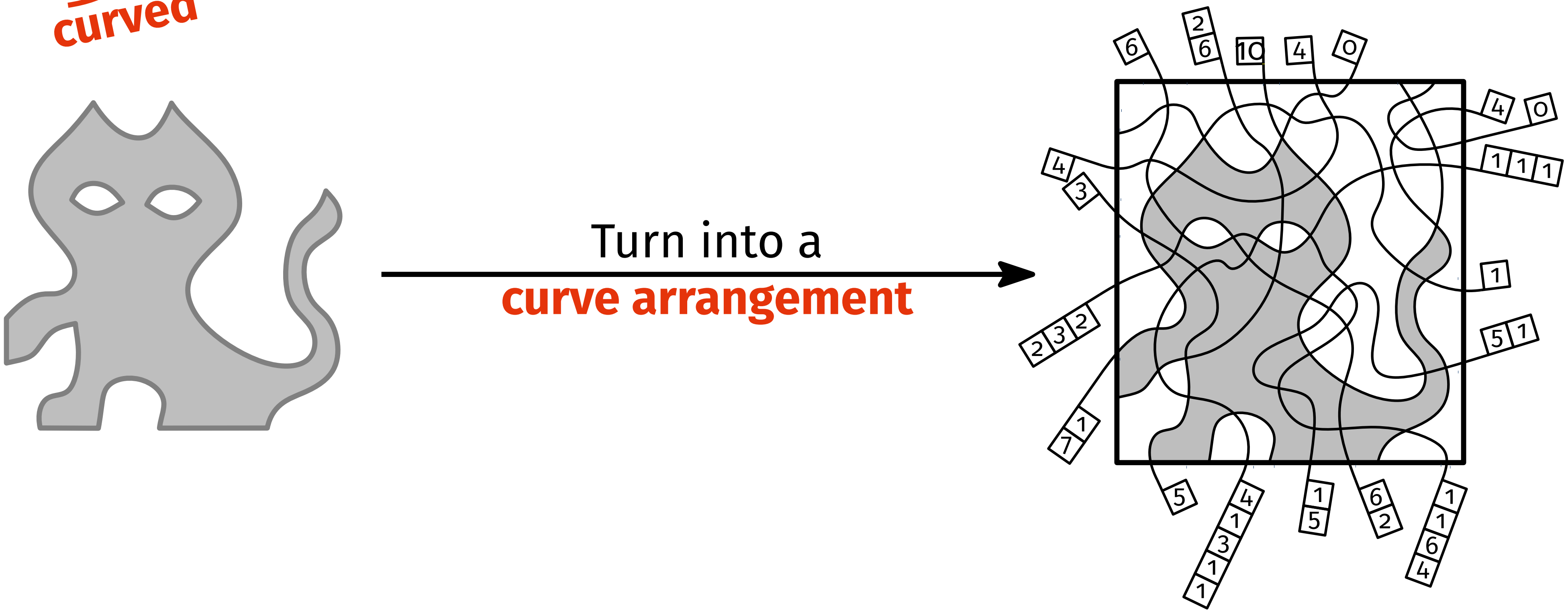
[Parment, ’15] [de Jong, ’16]

[van de Kerkhof, ’17] [van de Kerkhof et al., ’19]

Curved Nonograms

Given a shape…
curved

…compute a curved nonogram



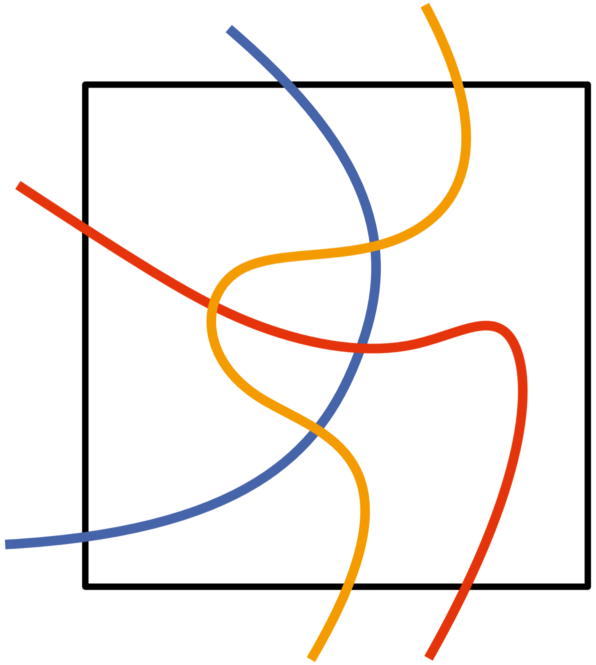
Turn into a
curve arrangement

Good: Previous work for generating nonograms
[Parment, ’15] [de Jong, ’16]
[van de Kerkhof, ’17] [van de Kerkhof et al., ’19]

Bad: Creates mostly **advanced** nonograms

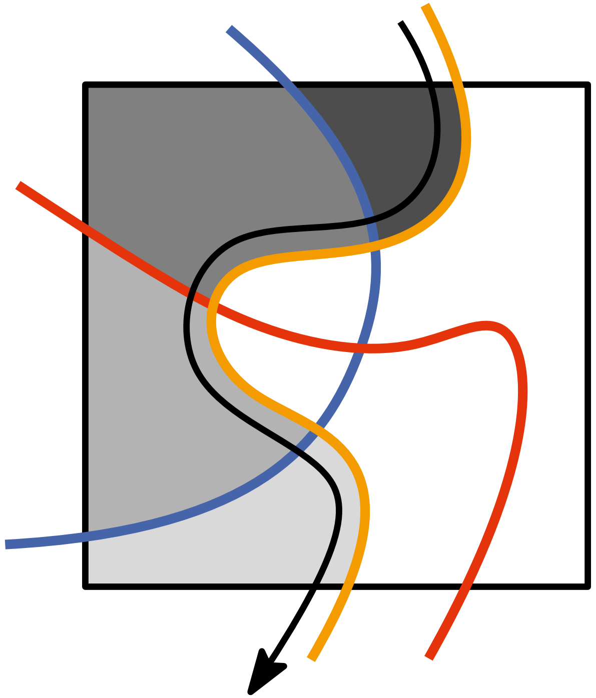
Types of Nonograms

Basic



Types of Nonograms

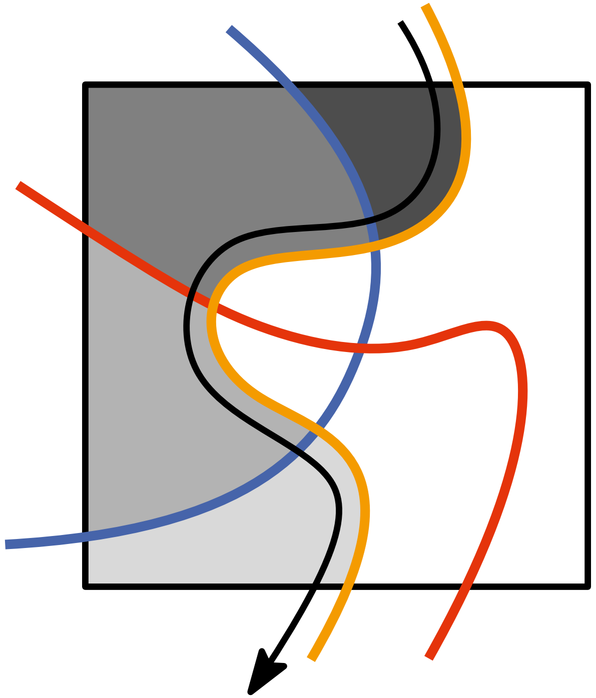
Basic



Unique faces along curves

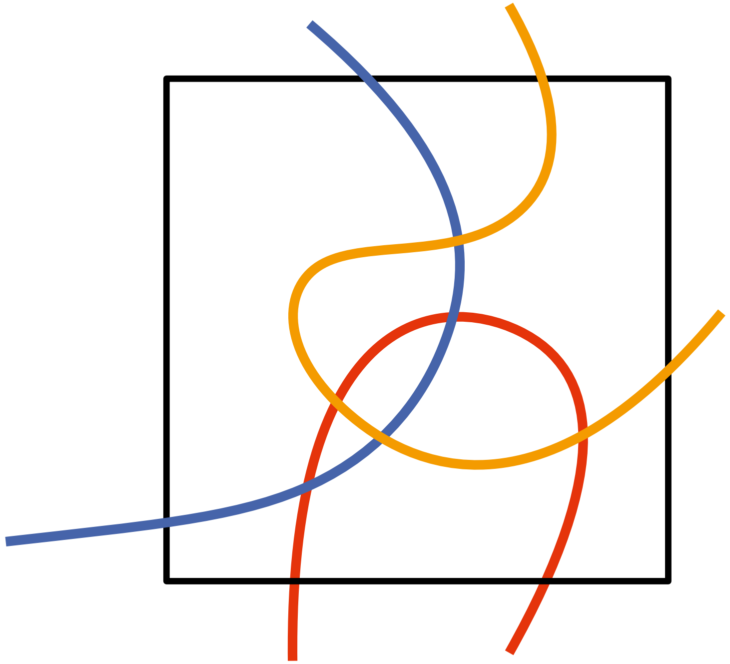
Types of Nonograms

Basic



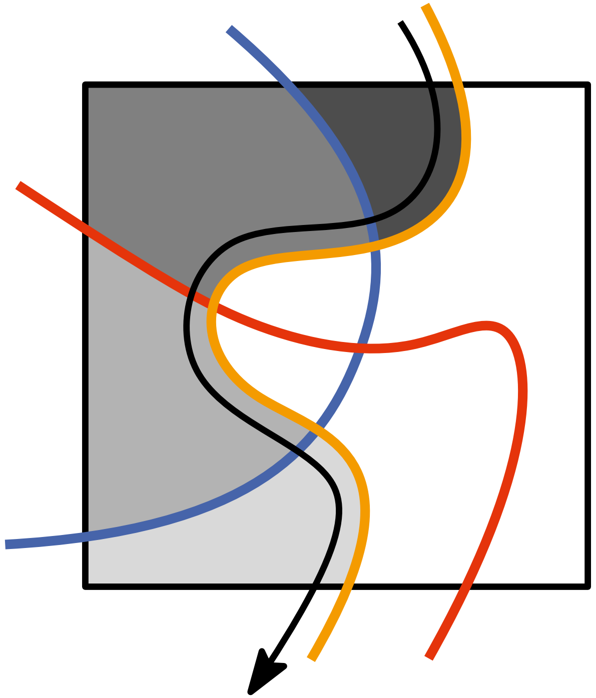
Unique faces
along curves

Advanced



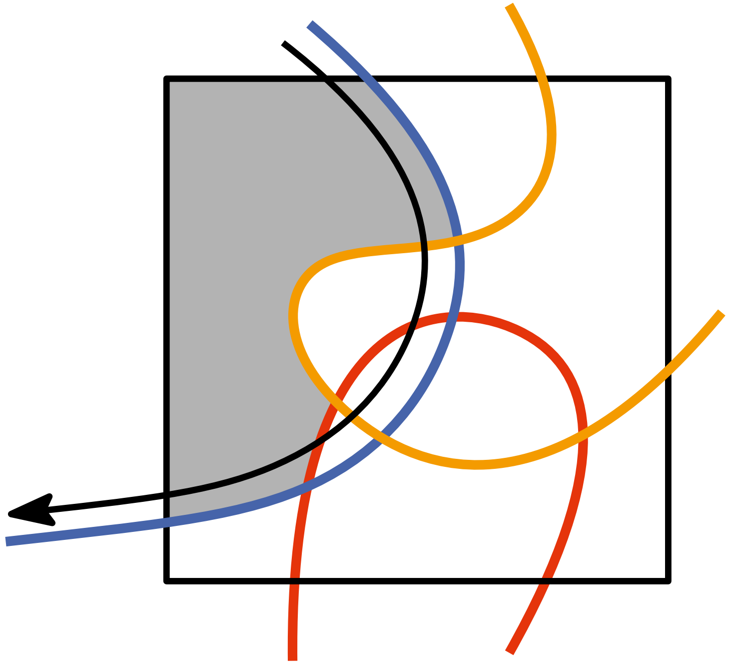
Types of Nonograms

Basic



Unique faces along curves

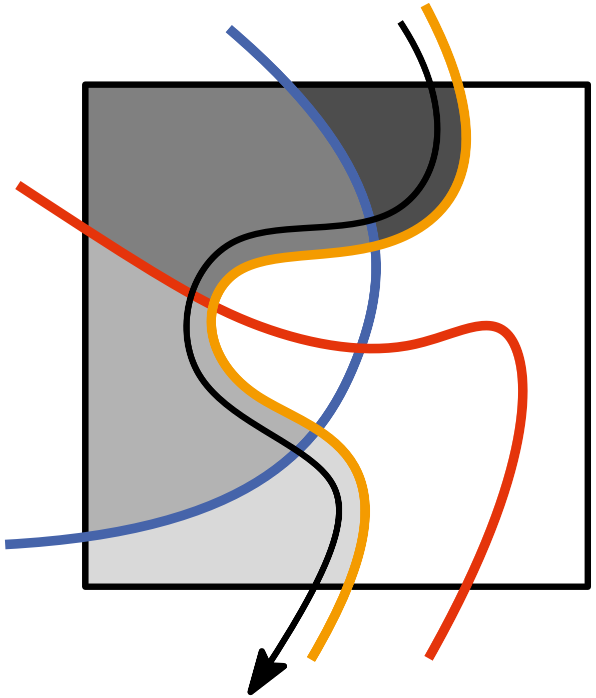
Advanced



Repeated (**popular**) faces along curves

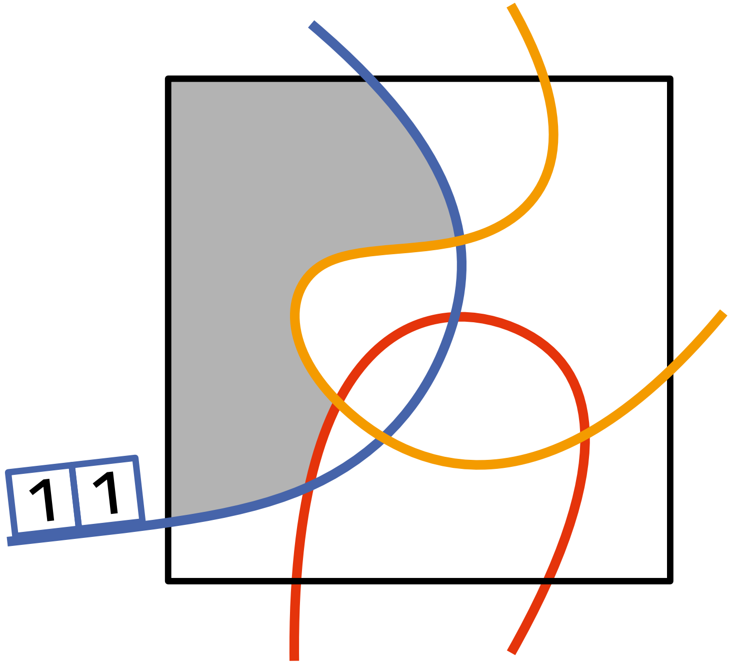
Types of Nonograms

Basic



Unique faces
along curves

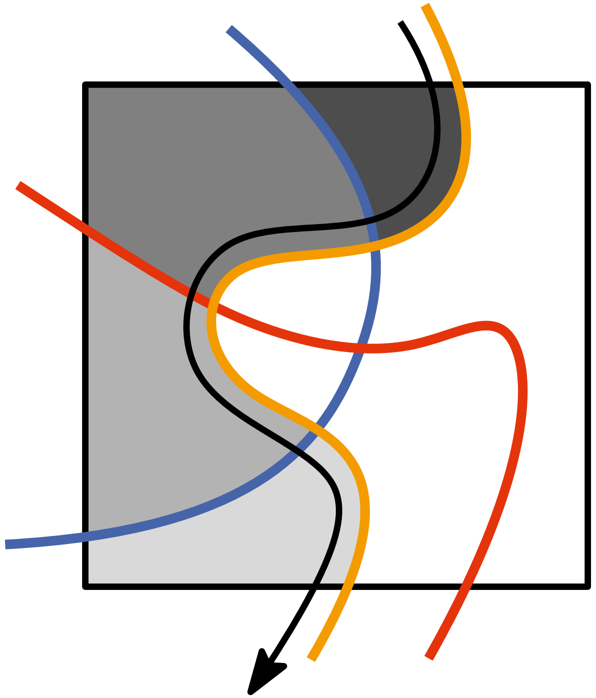
Advanced



Repeated (**popular**) faces
along curves

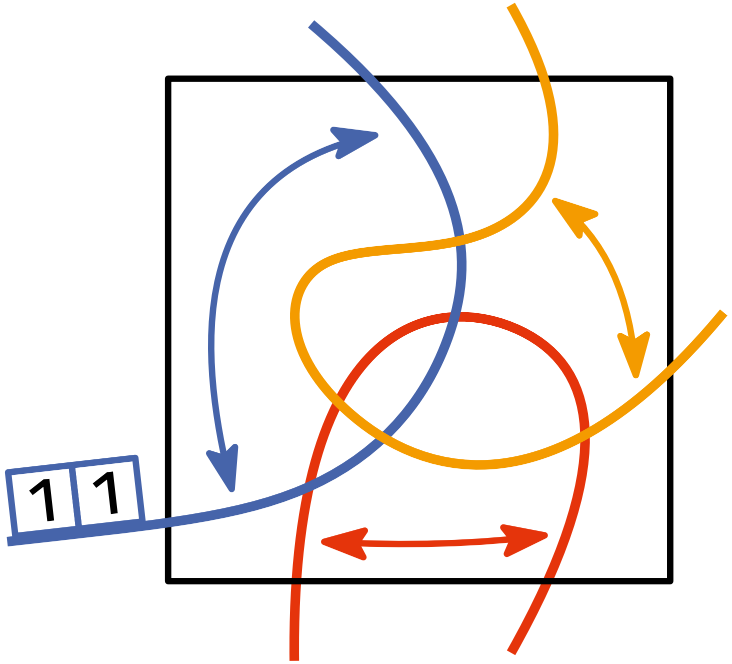
Types of Nonograms

Basic



Unique faces
along curves

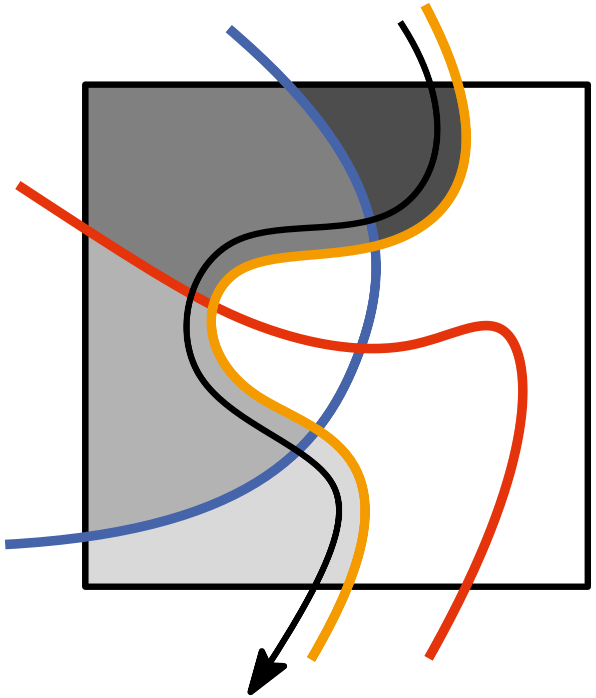
Advanced



Repeated (**popular**) faces
along curves

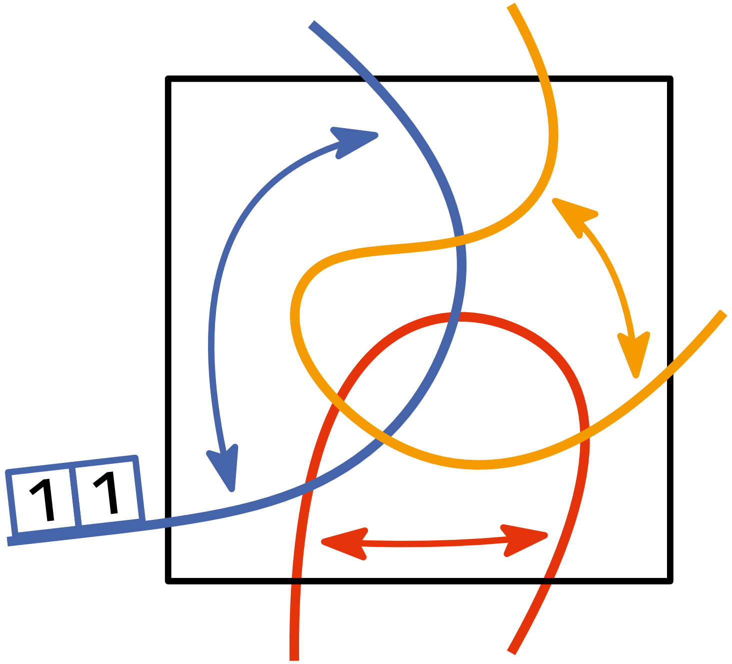
Types of Nonograms

Basic

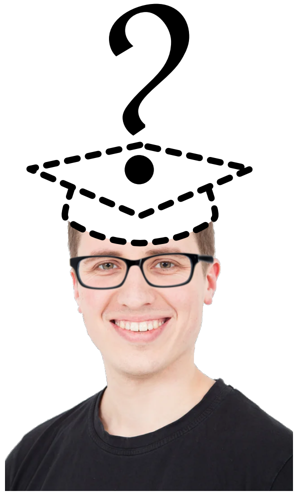


Unique faces along curves

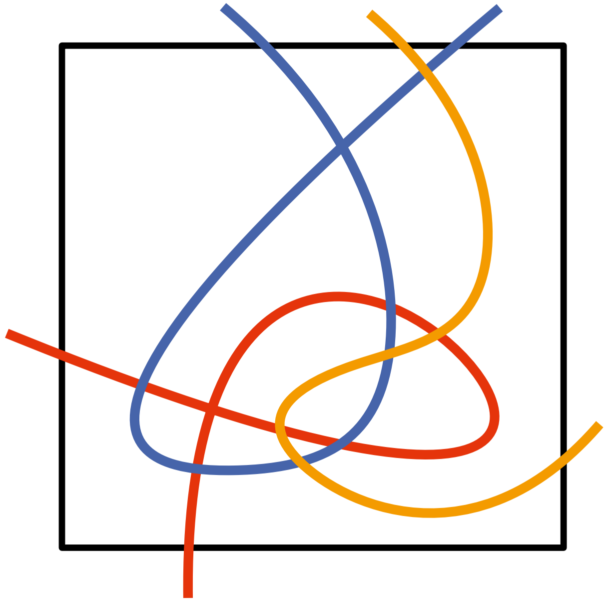
Advanced



Repeated (**popular**) faces along curves



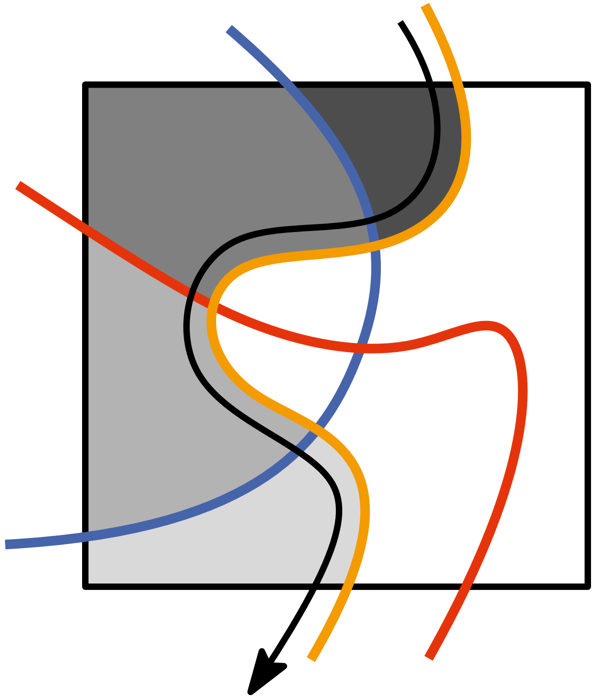
Expert



Self intersections

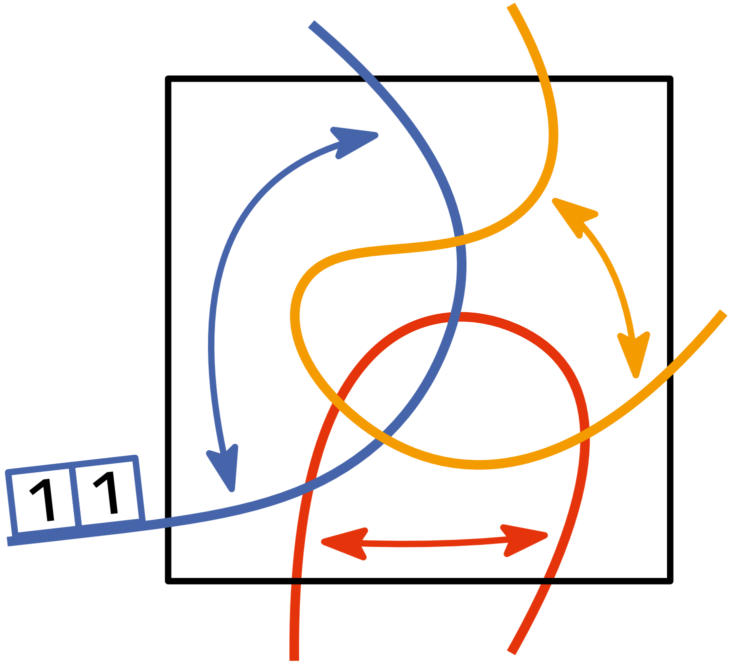
Types of Nonograms

Basic



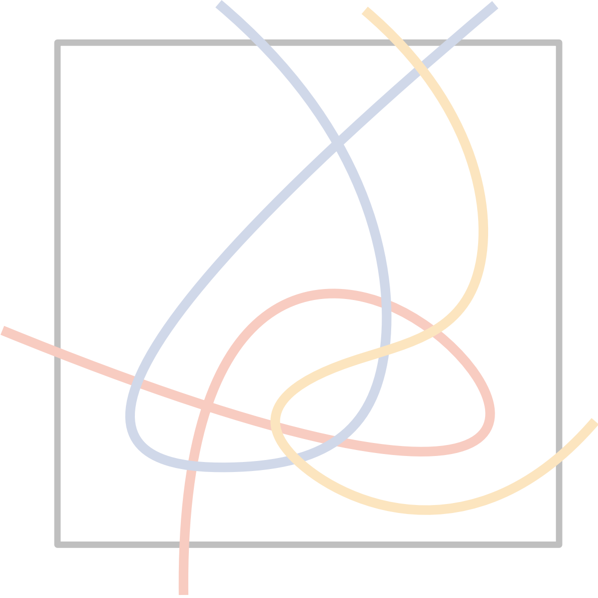
Unique faces
along curves

Advanced



Repeated (**popular**) faces
along curves

Expert

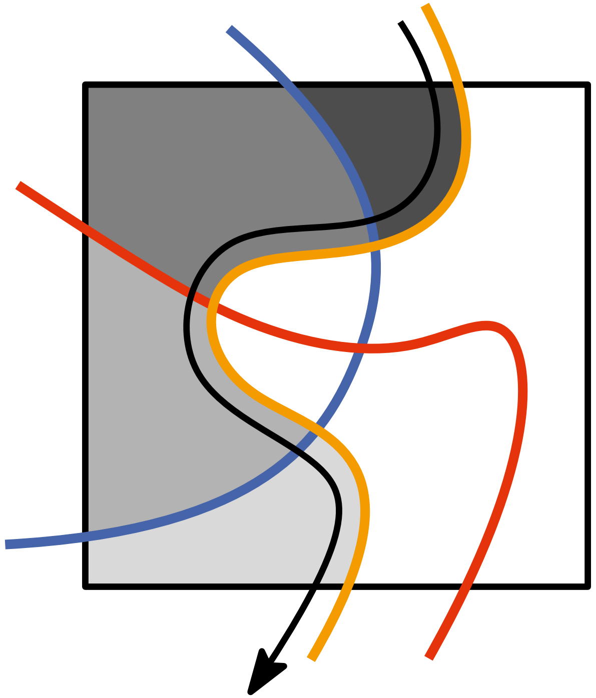


Self intersections

Types of Nonograms

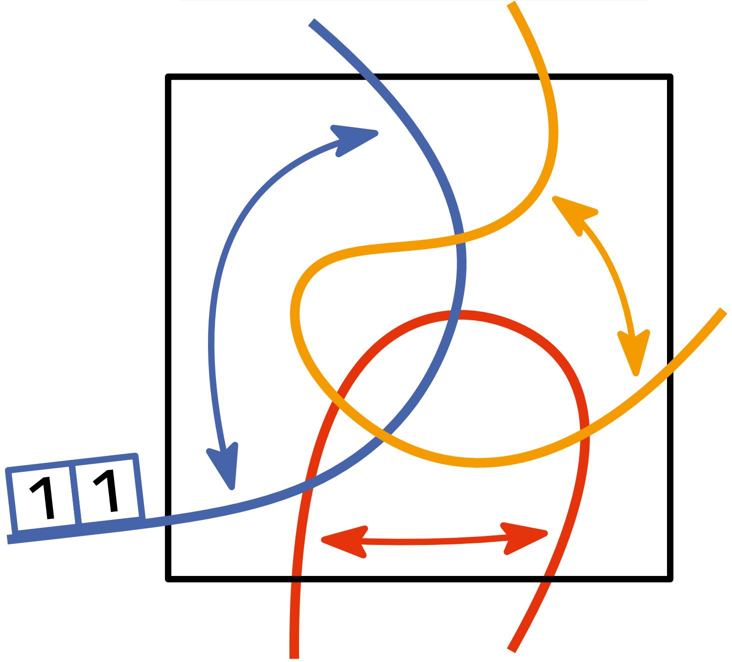
[van de Kerkhof et al., ’19]

Basic



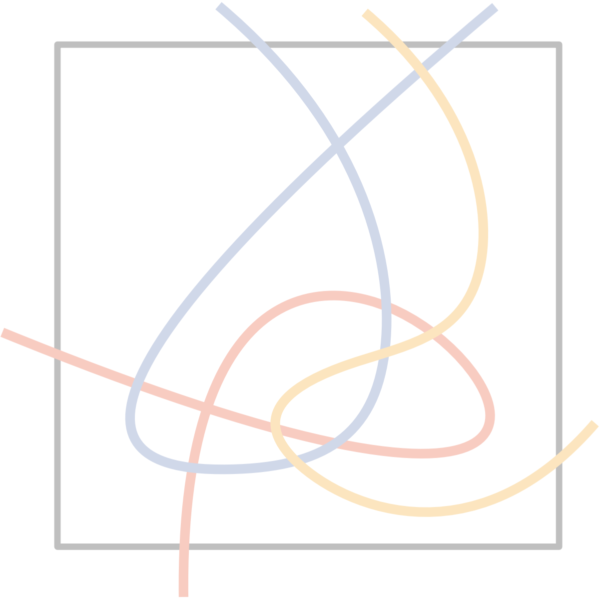
Unique faces along curves

Advanced



Repeated (**popular**) faces along curves

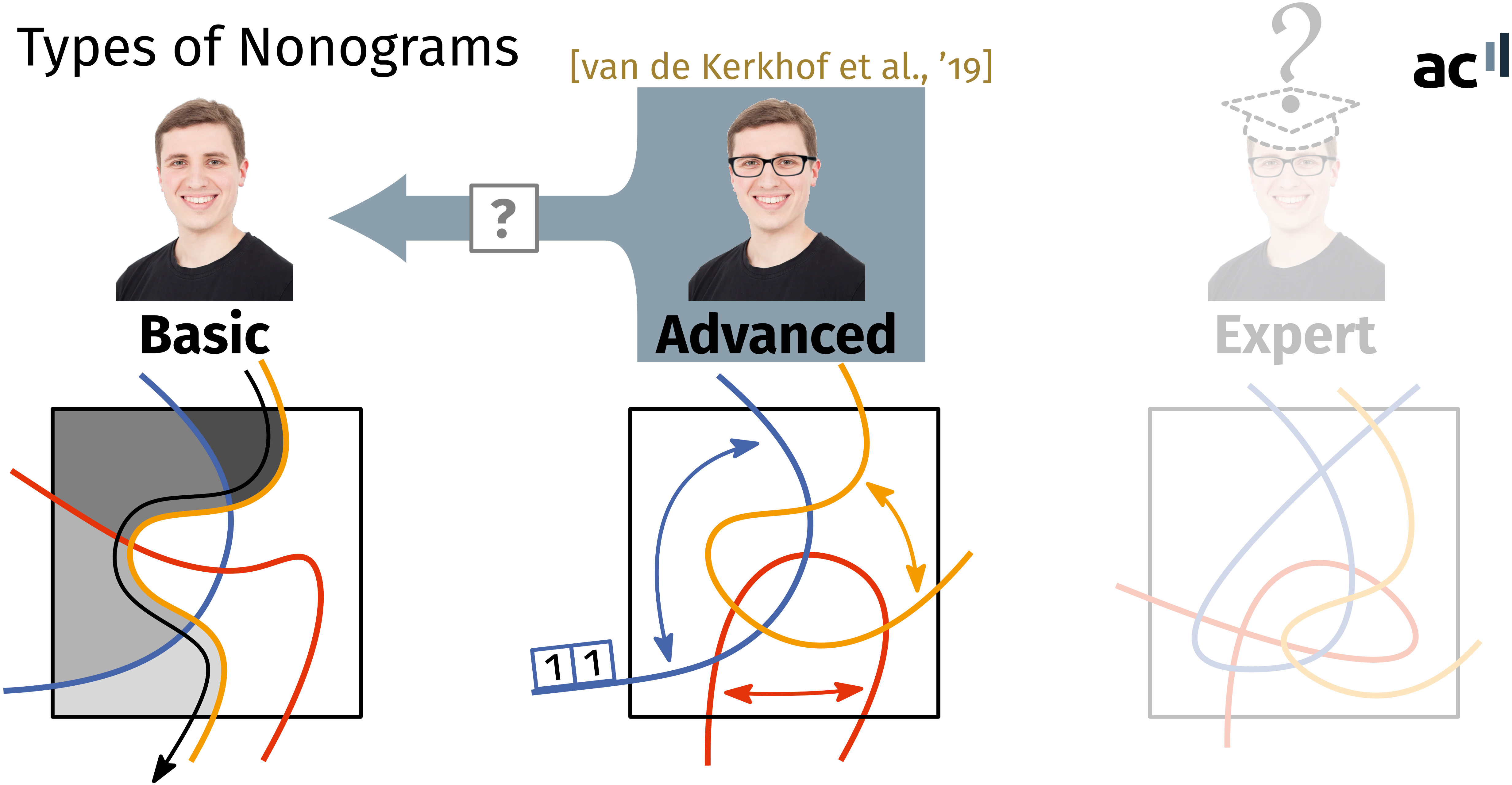
Expert



Self intersections

Types of Nonograms

[van de Kerkhof et al., ’19]



Basic

Advanced

Expert

Unique faces along curves

Repeated (**popular**) faces along curves

Self intersections

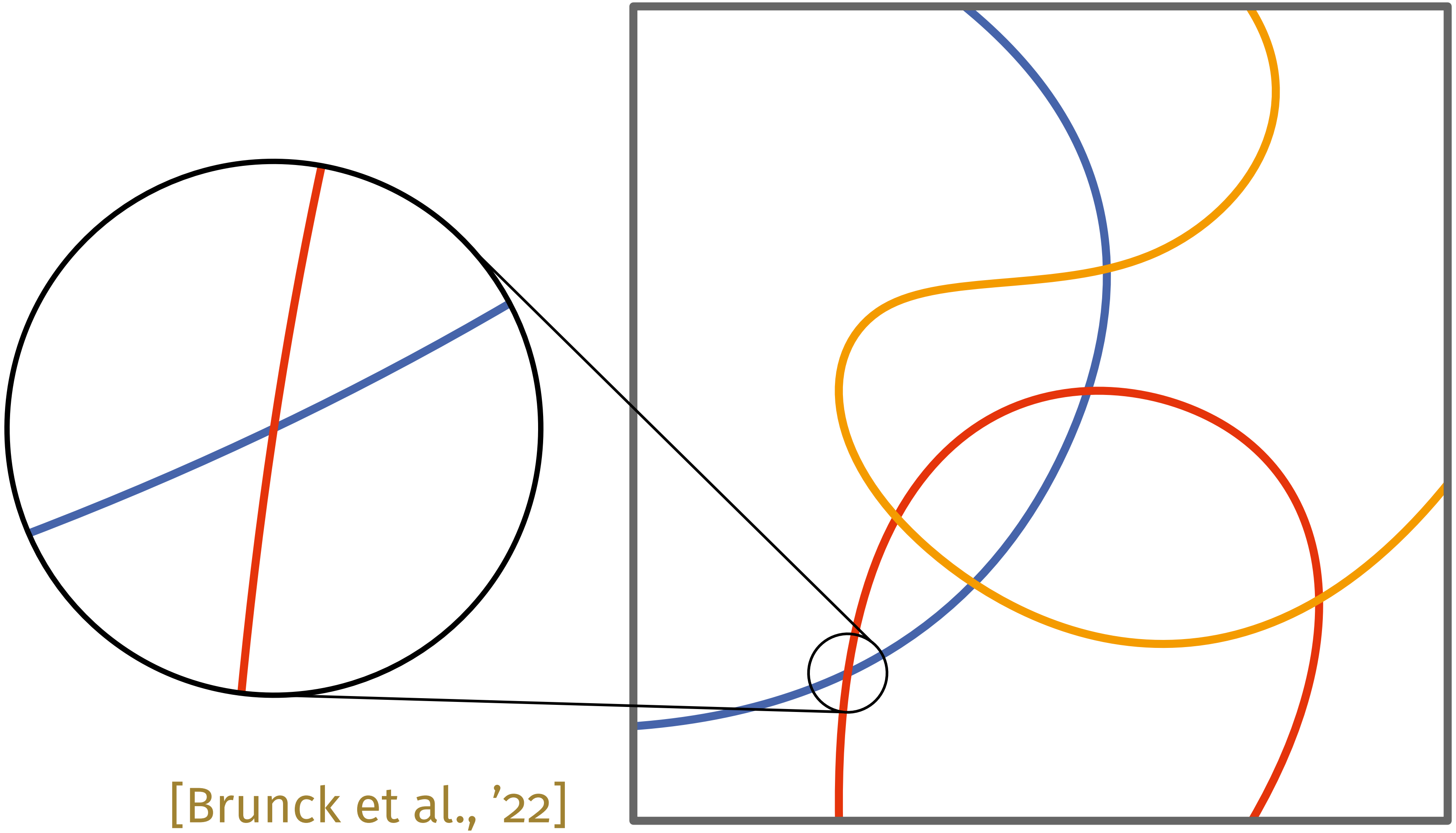
Outline

Nonograms

How to remove popular faces

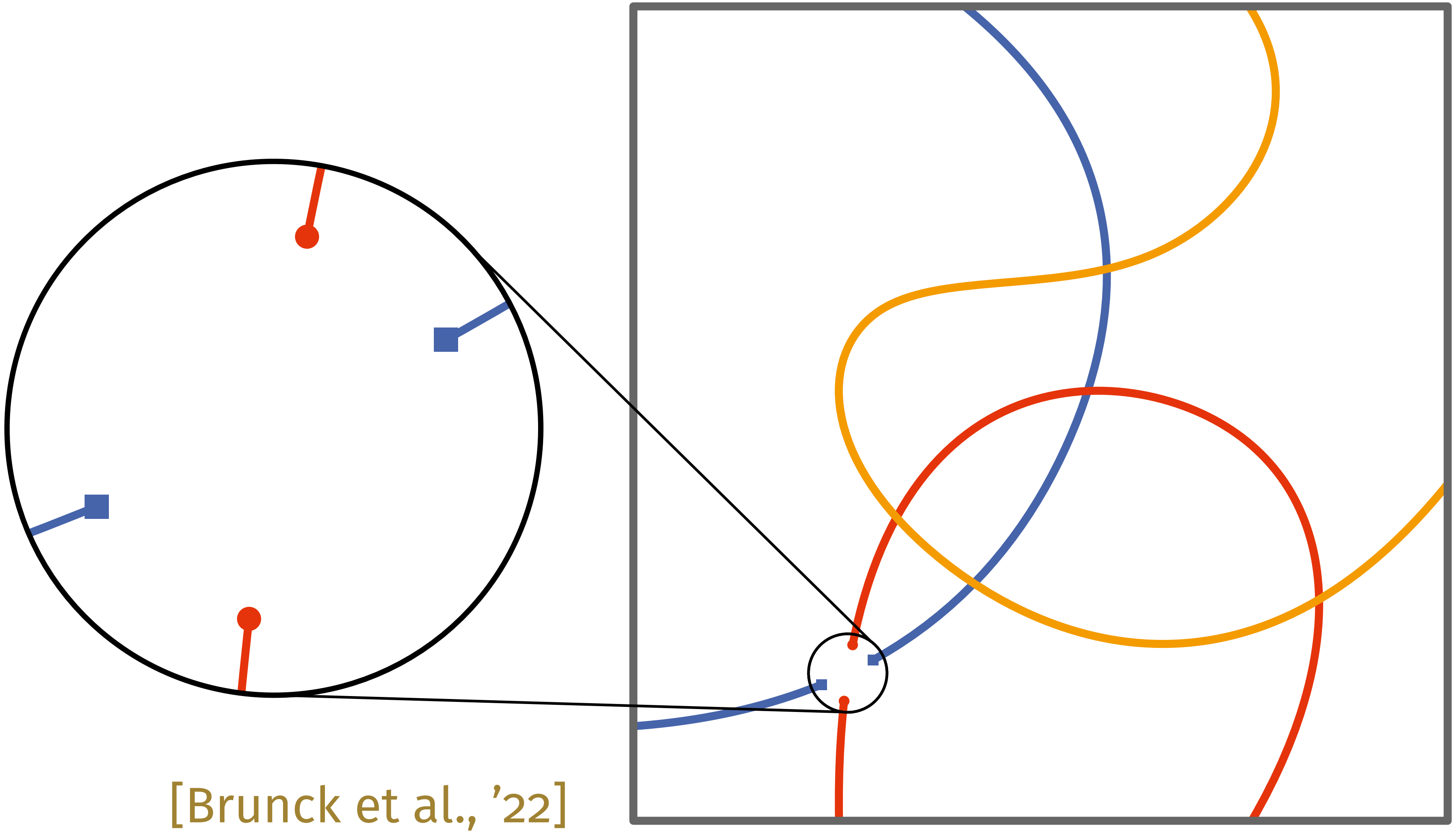
Removing Popular Faces

Removing Popular Faces



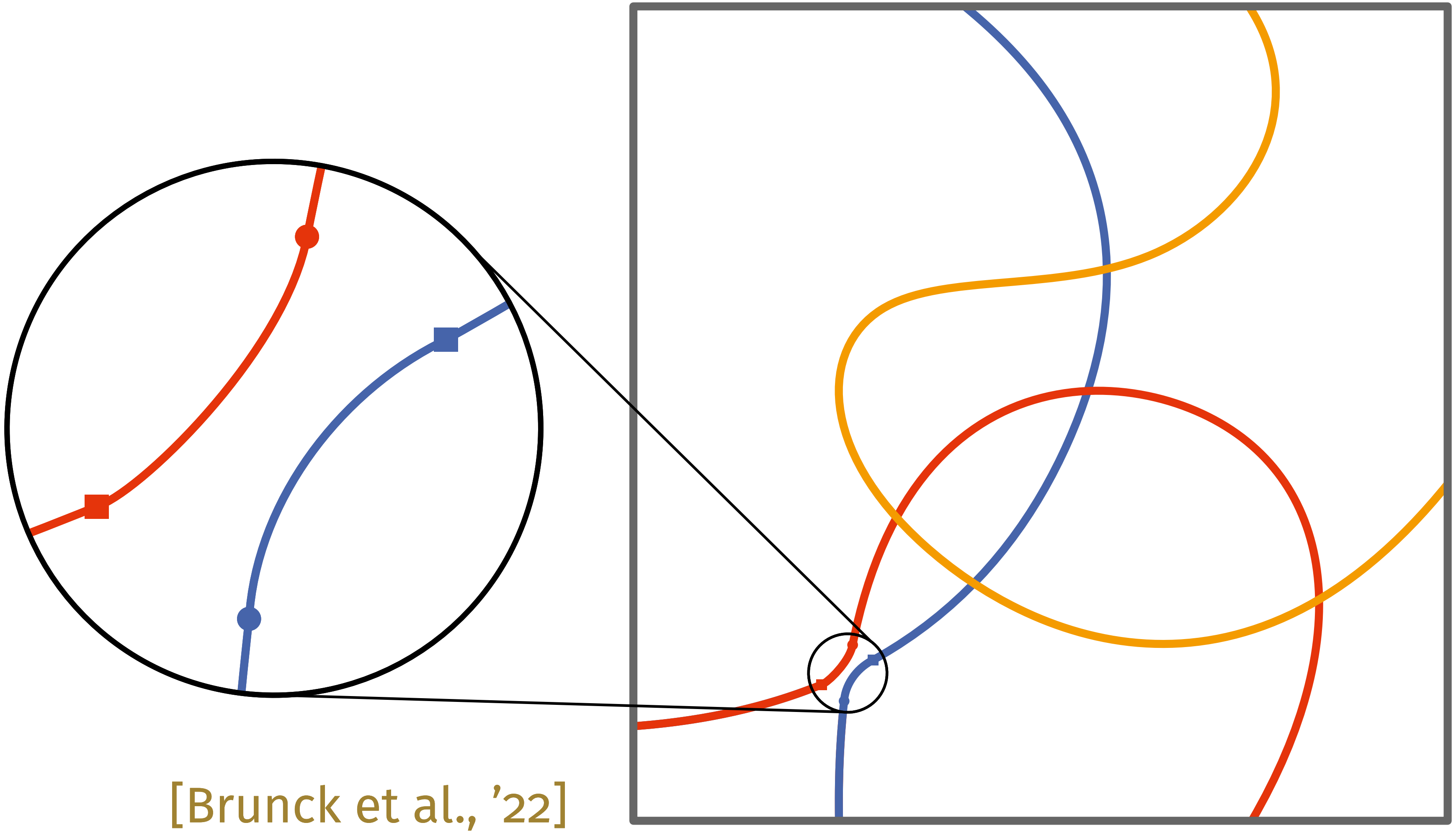
[Brunck et al., ’22]

Removing Popular Faces



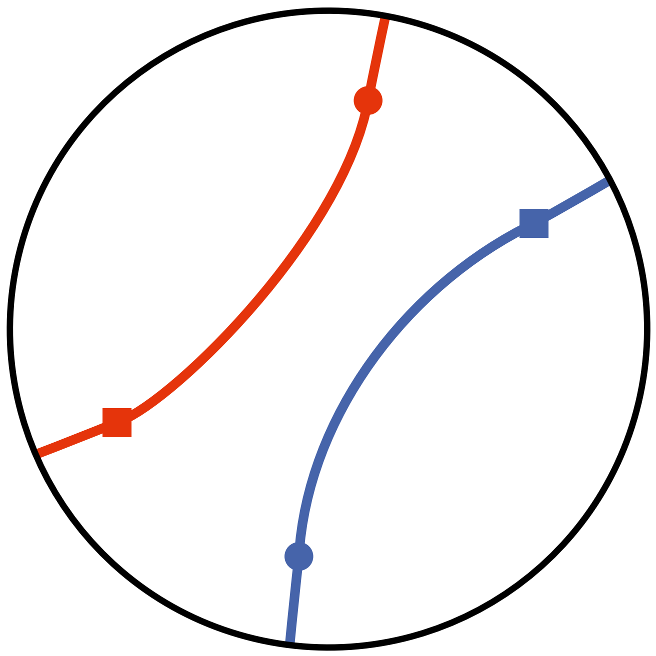
[Brunck et al., ’22]

Removing Popular Faces

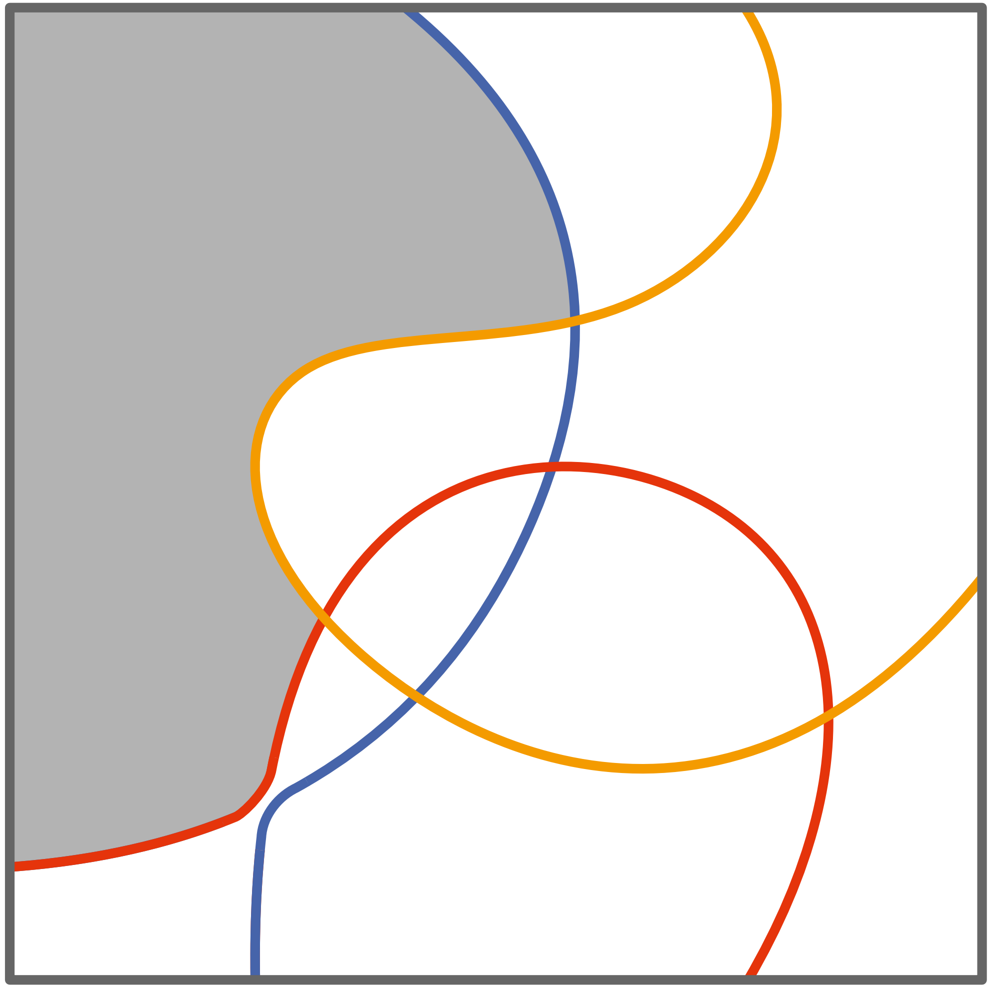


[Brunck et al., '22]

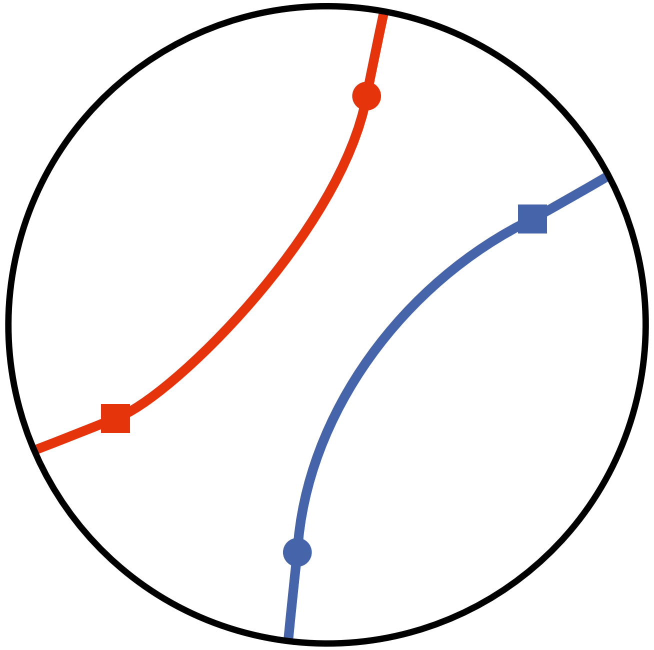
Removing Popular Faces



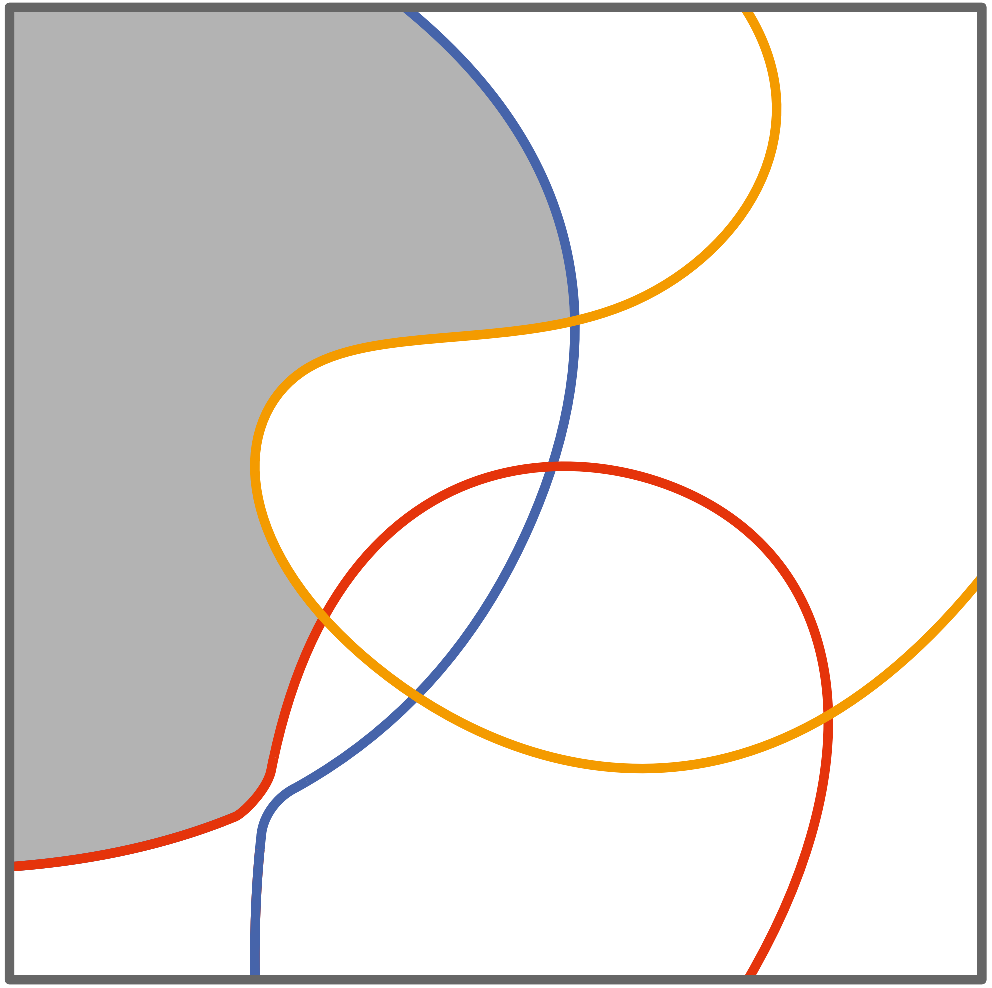
[Brunck et al., ’22]

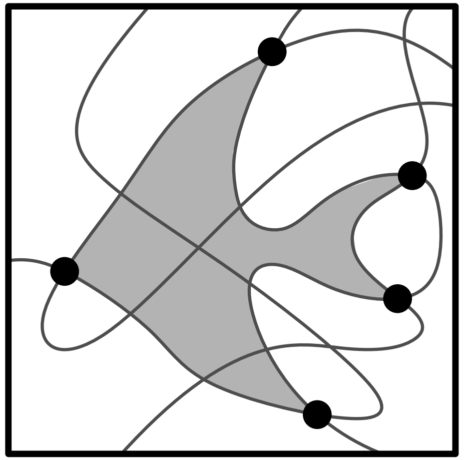


Removing Popular Faces



[Brunck et al., '22]

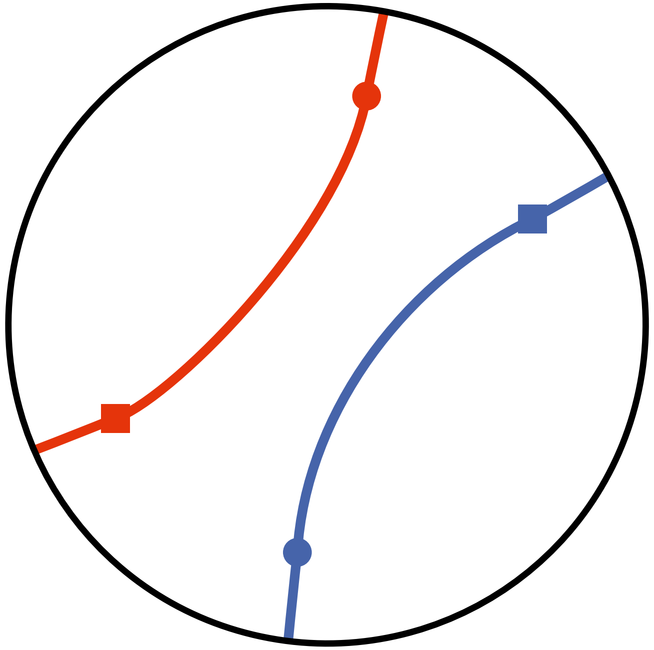




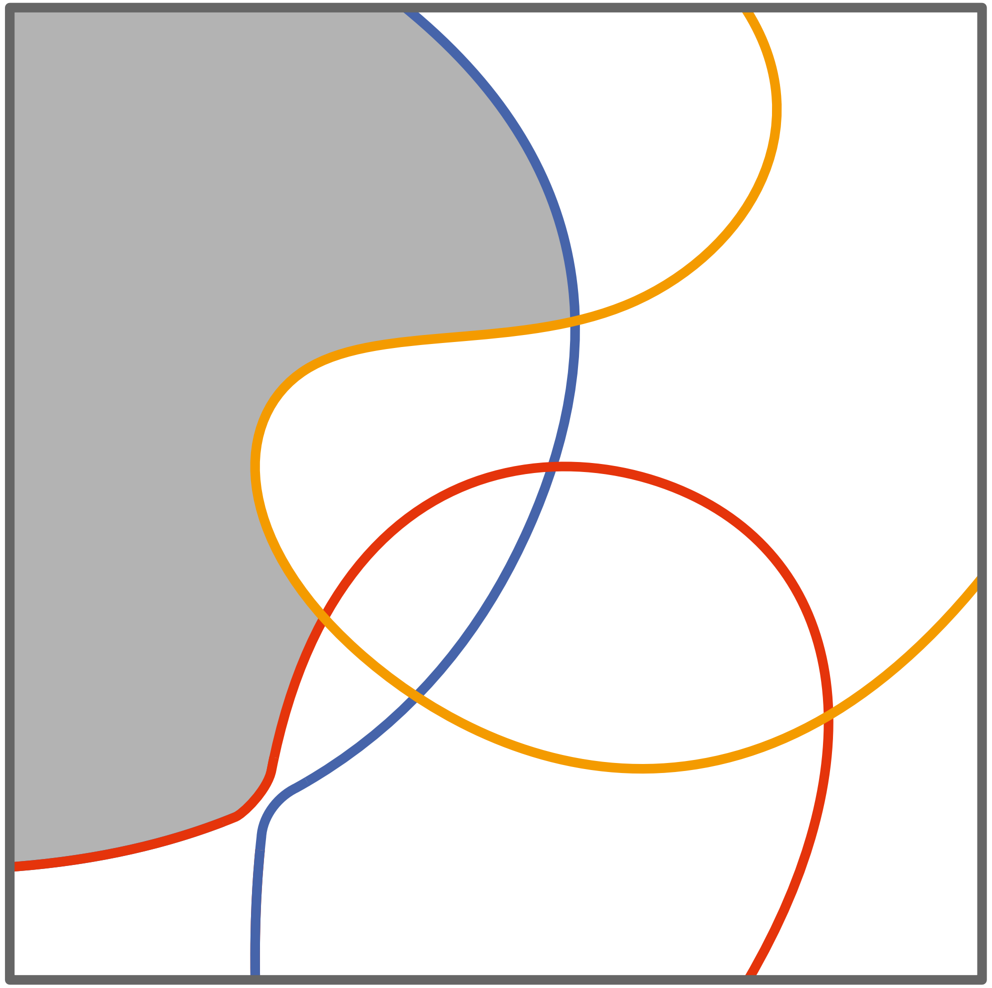
[van de Kerkhof et al., '19]

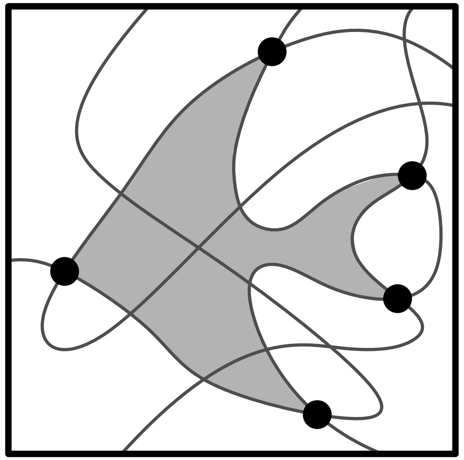
Removing Popular Faces

We do not want to change existing geometry!



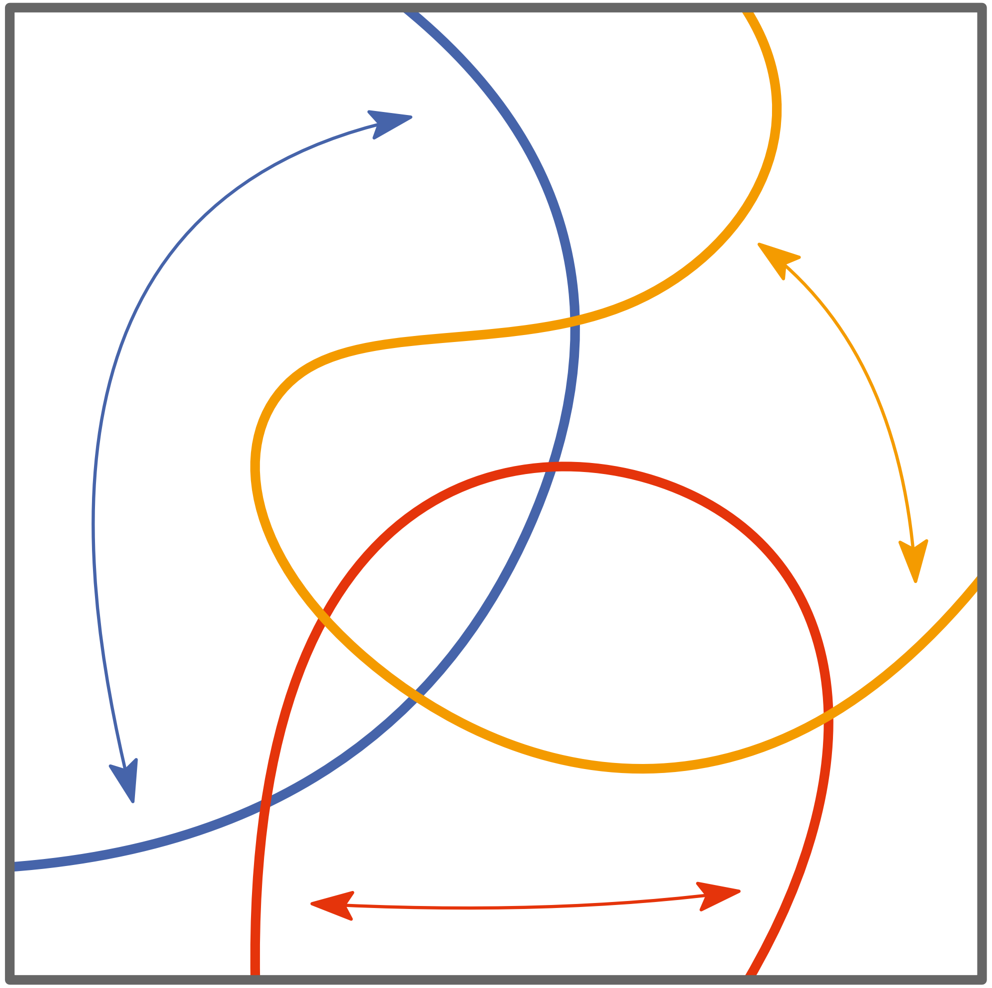
[Brunck et al., ’22]



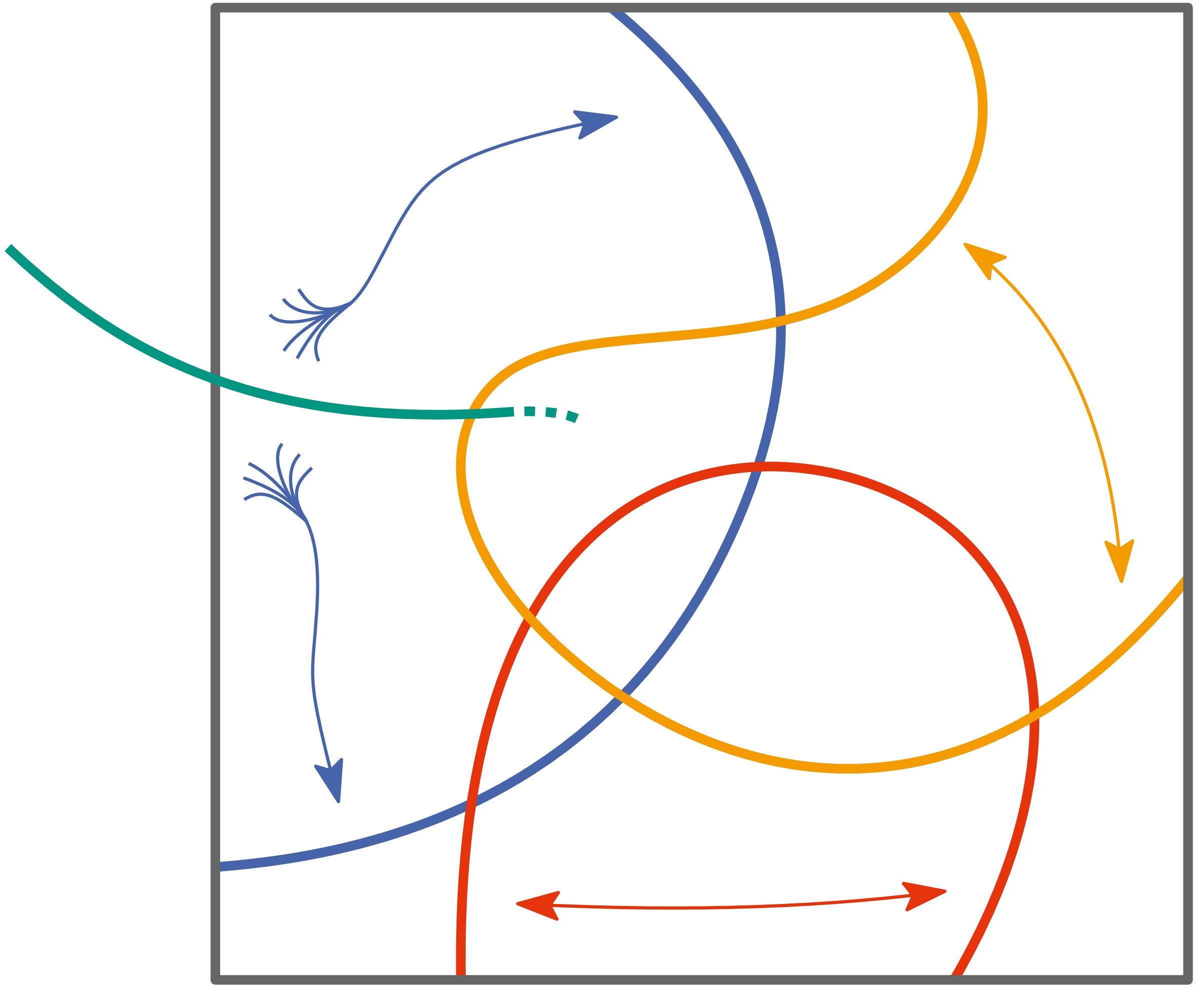


[van de Kerkhof et al., ’19]

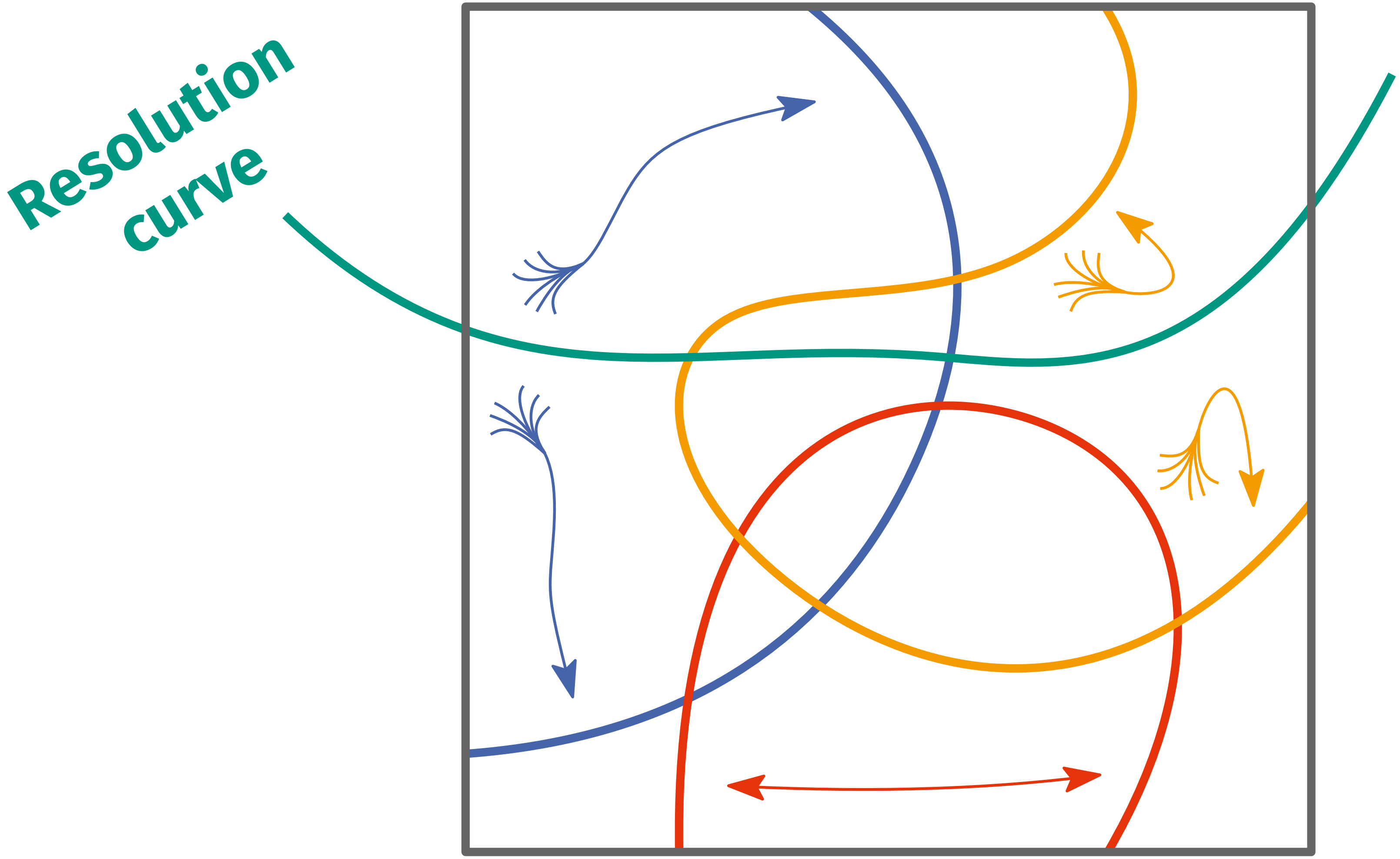
Removing Popular Faces



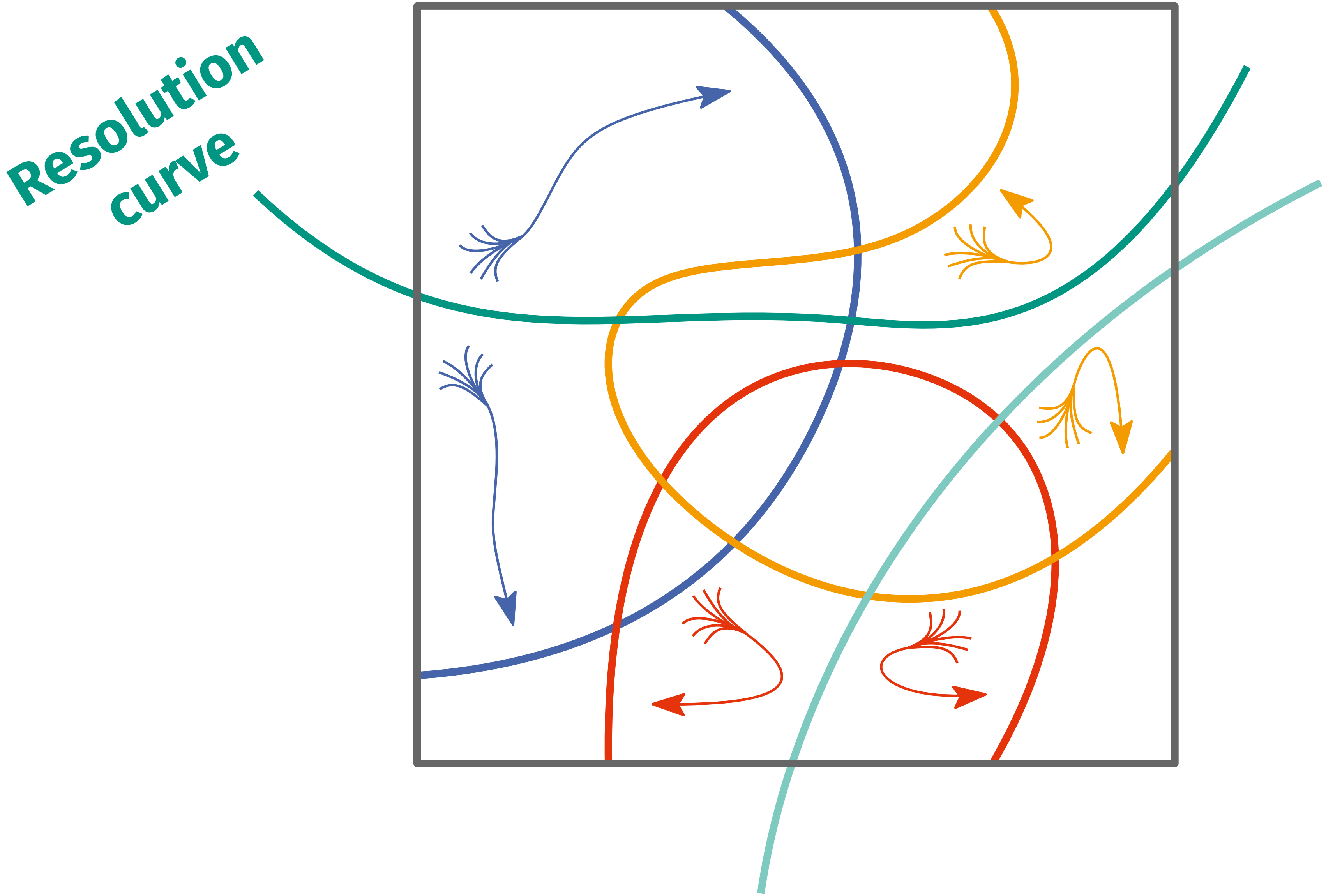
Removing Popular Faces



Removing Popular Faces

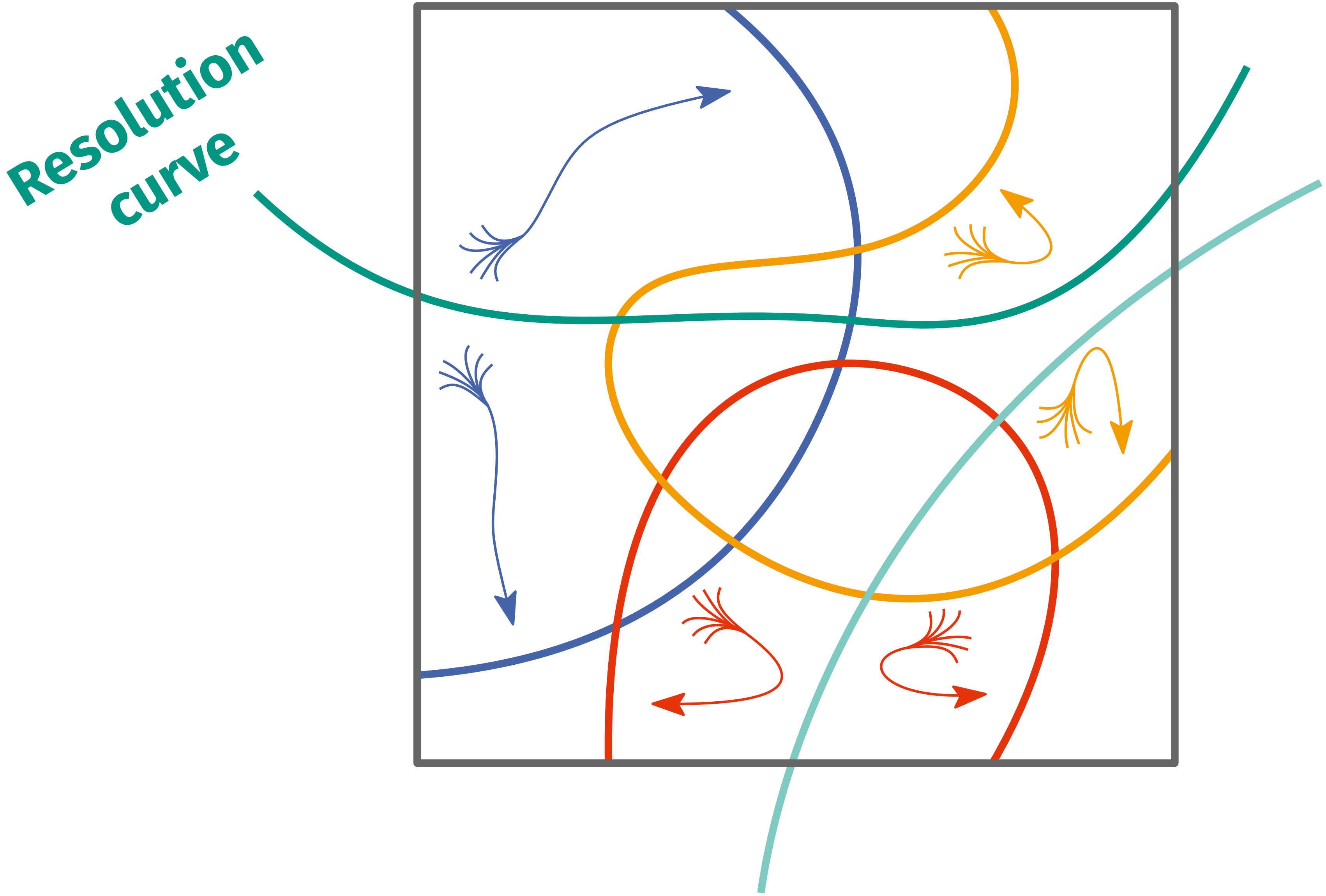


Removing Popular Faces

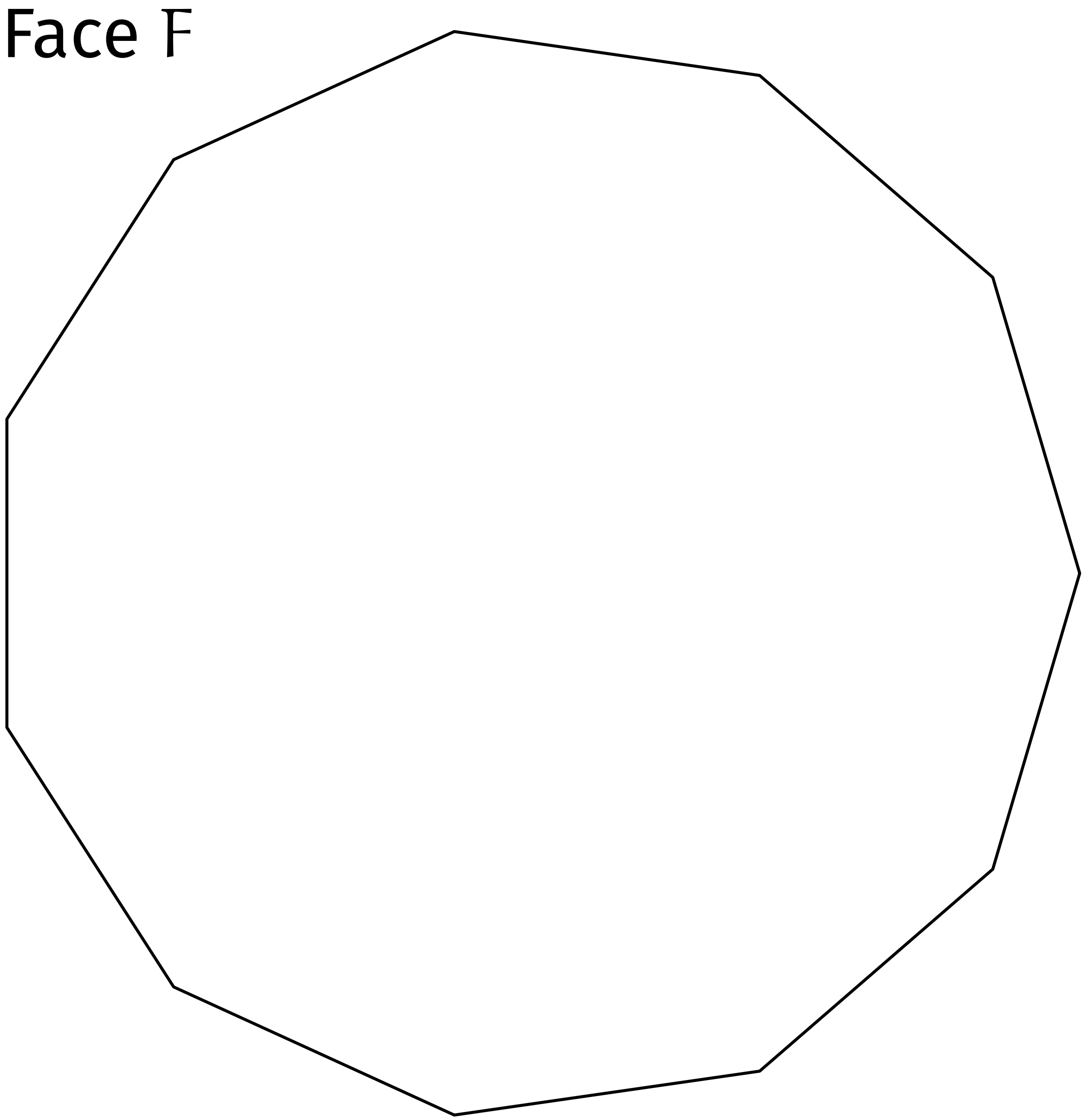


Removing Popular Faces

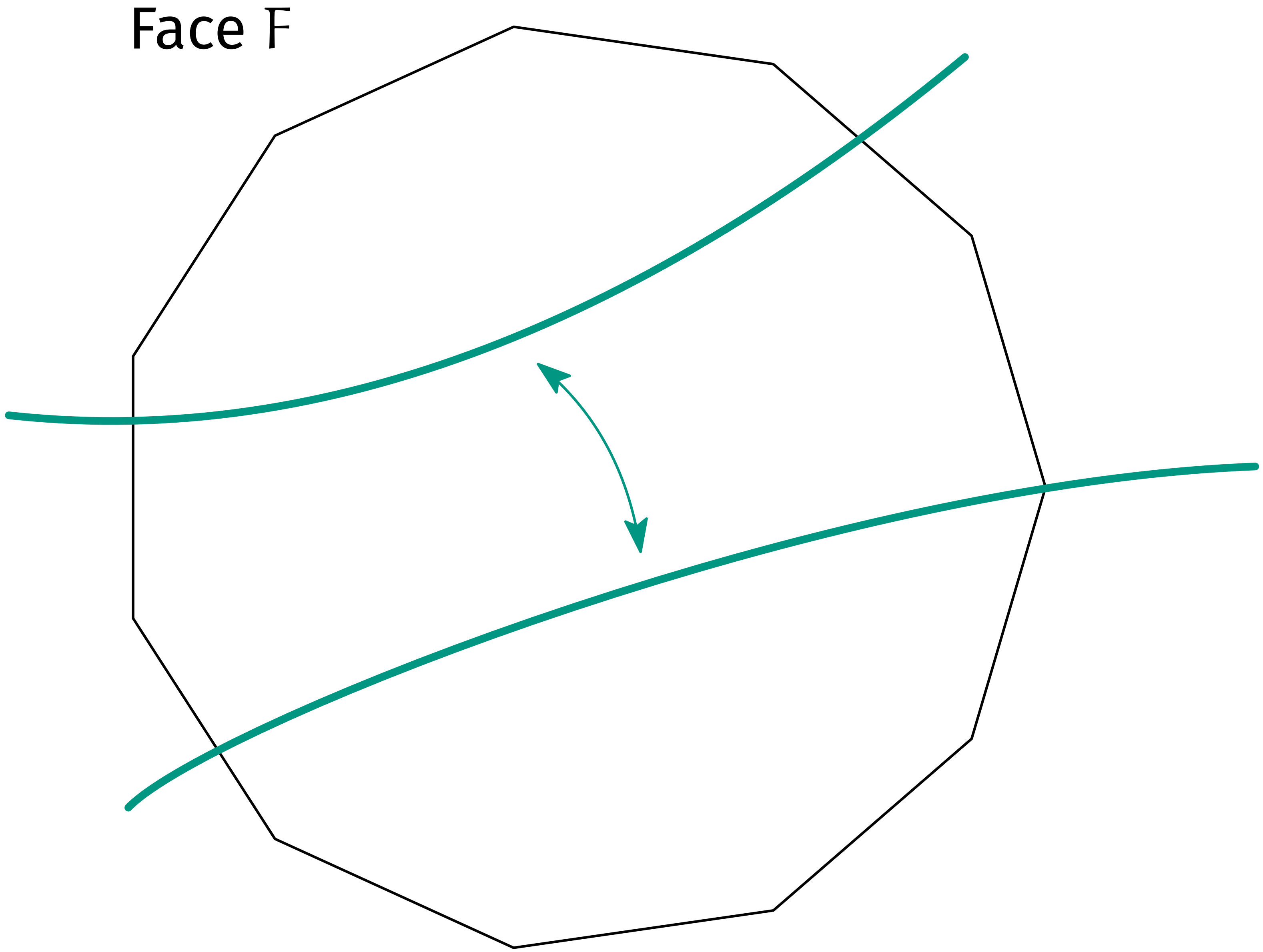
Can we do it with just 1 resolution curve?



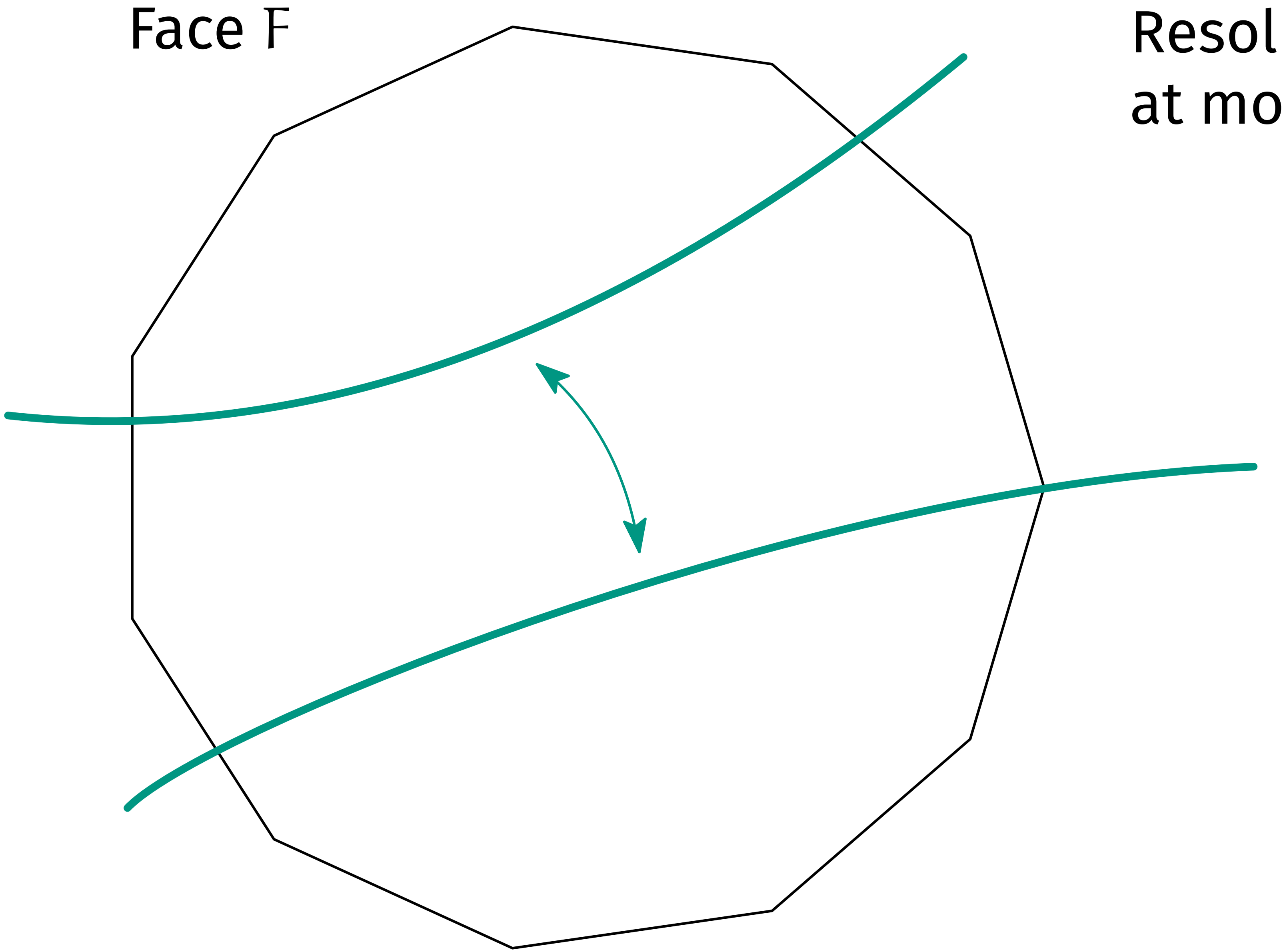
Resolving a Single Face with One Curve



Resolving a Single Face with One Curve

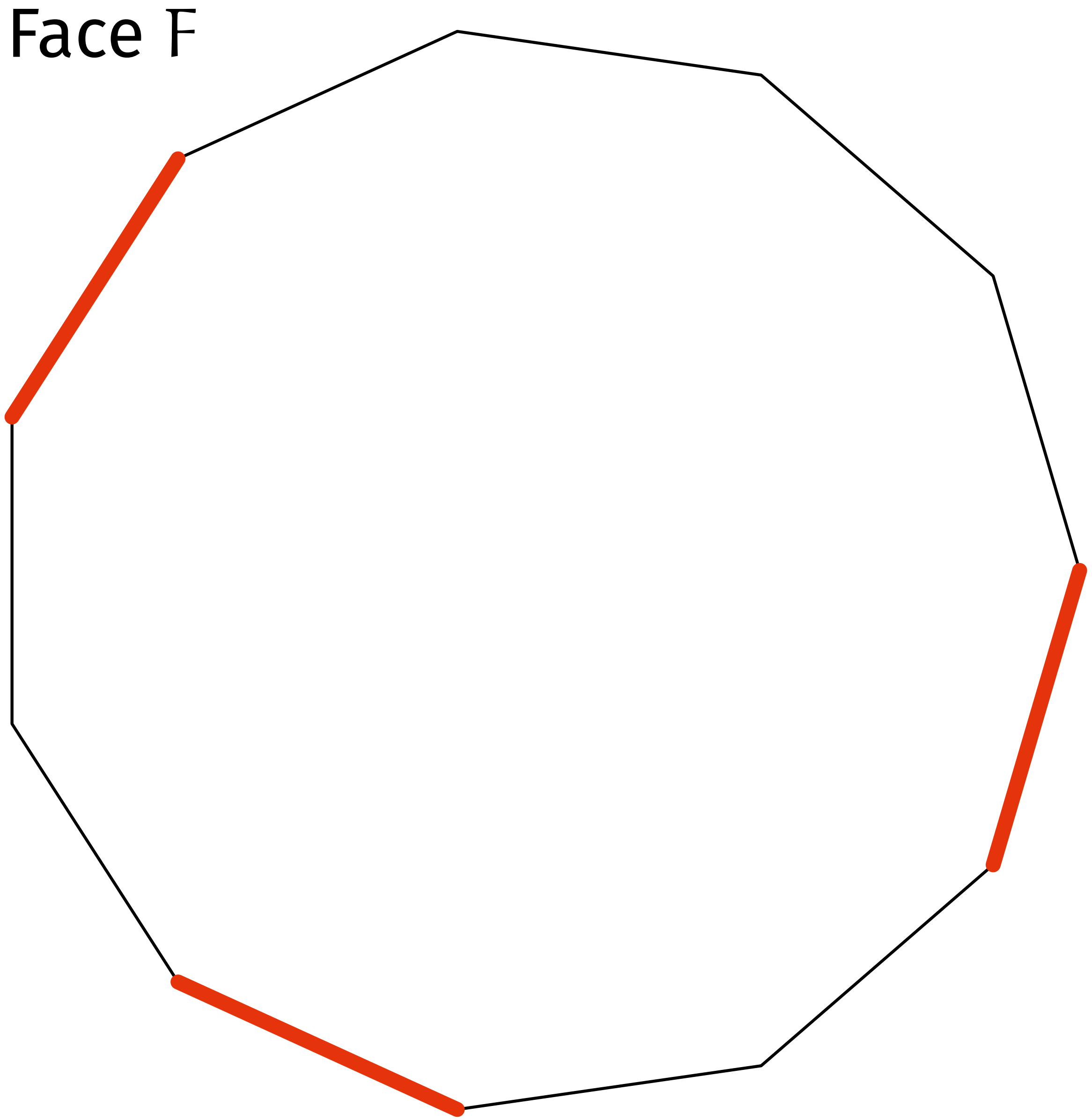


Resolving a Single Face with One Curve



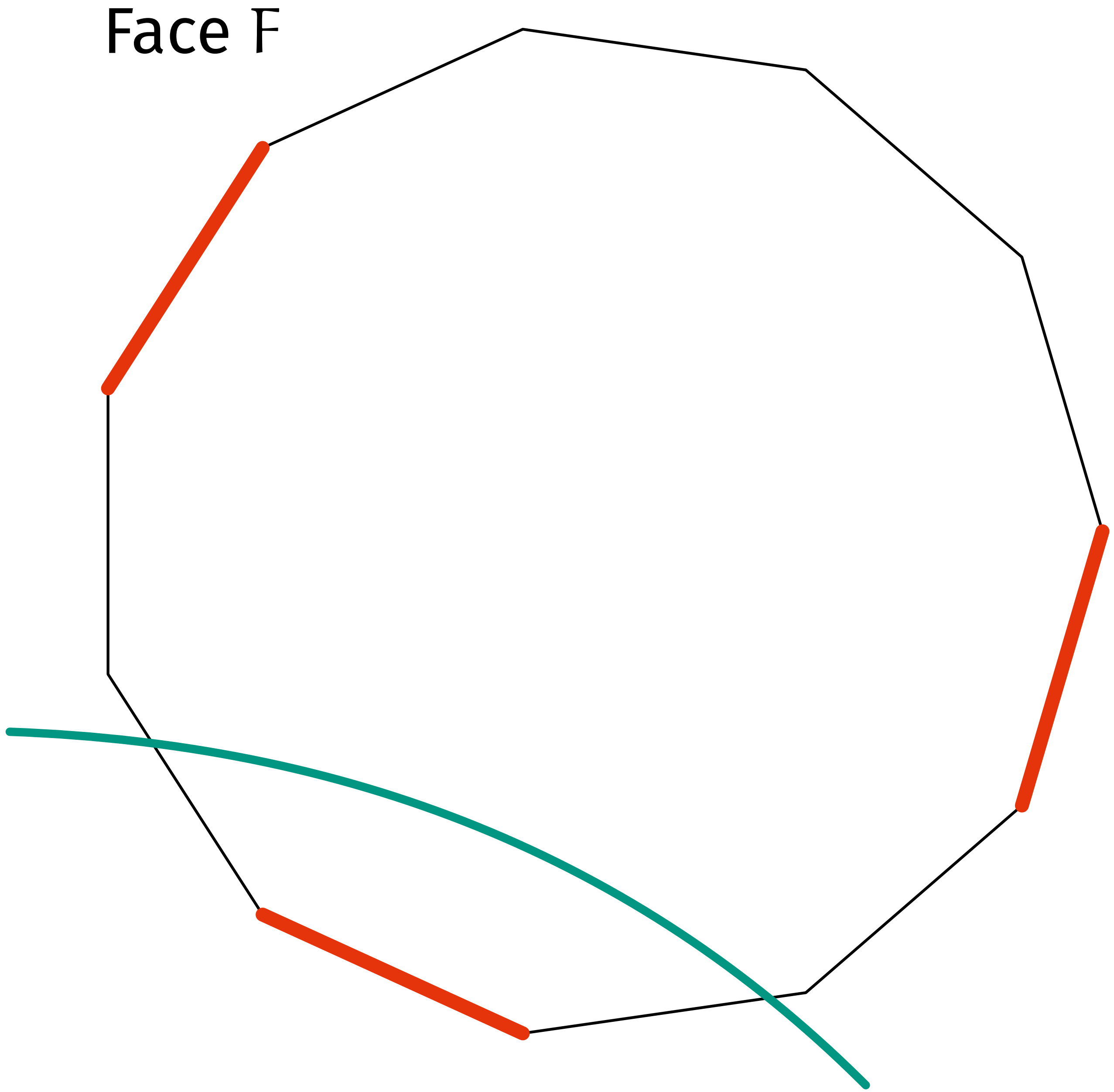
Resolution curves cross faces at most one time

Resolving a Single Face with One Curve



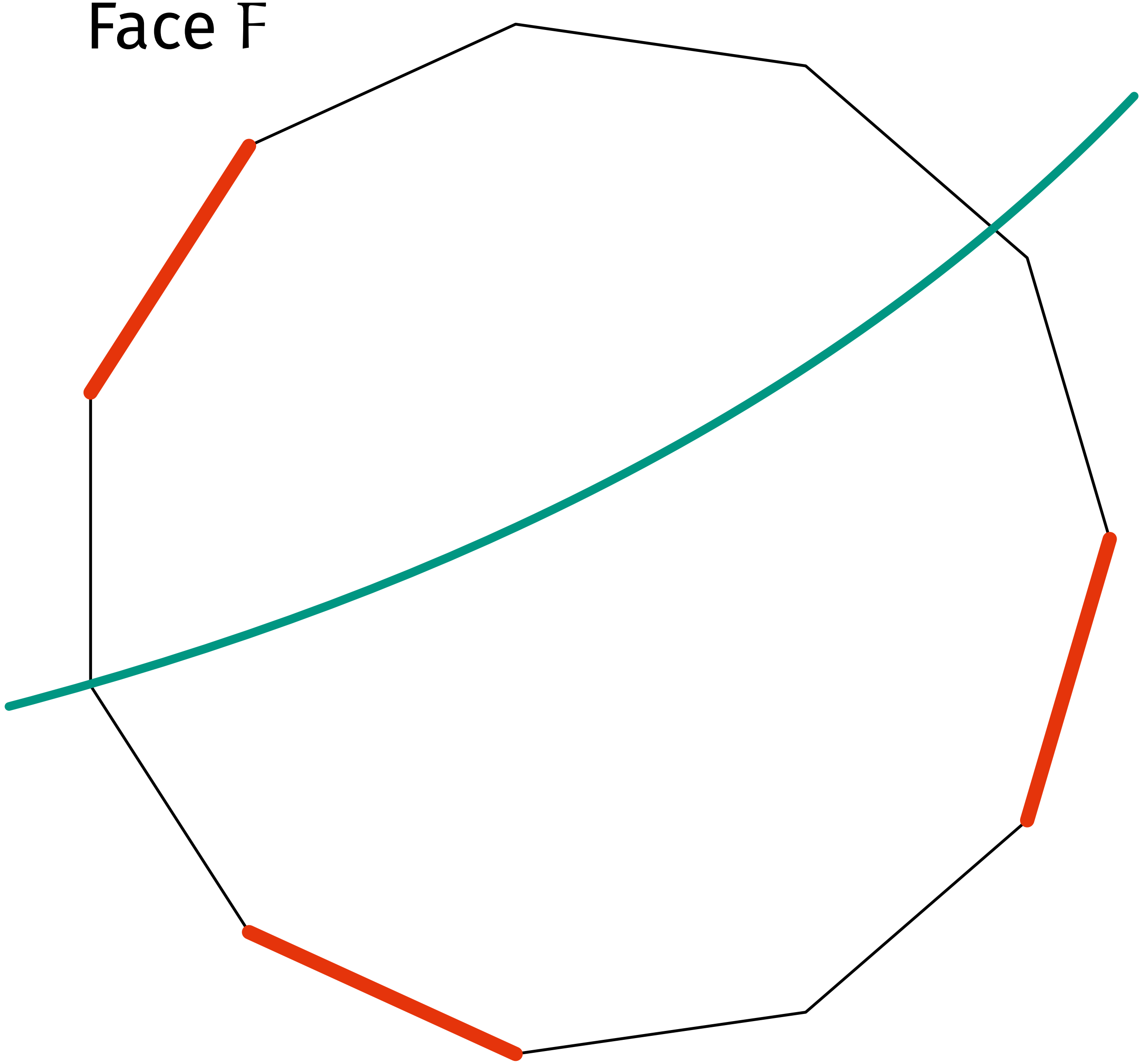
Resolution curves cross faces at most one time

Resolving a Single Face with One Curve



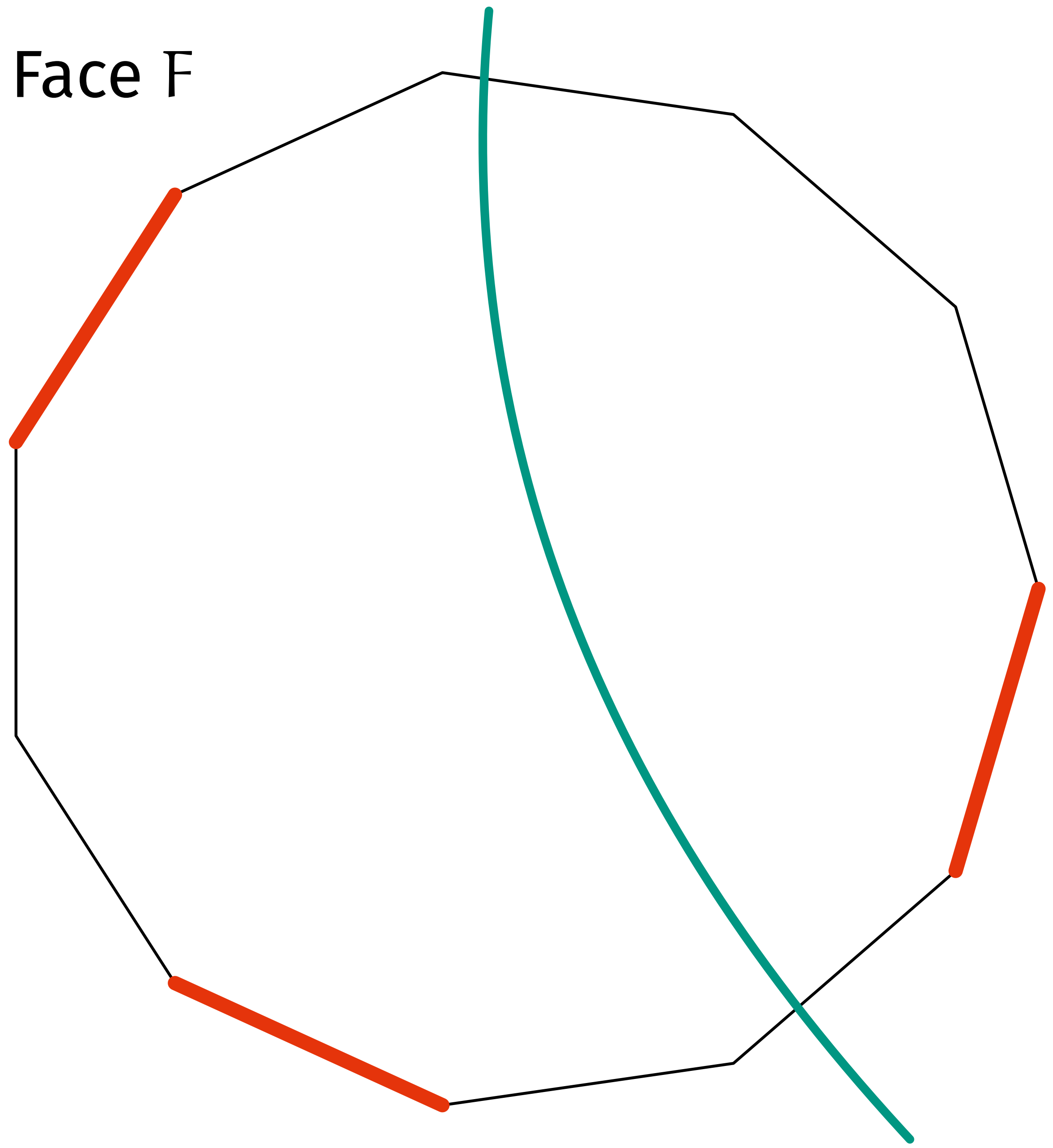
Resolution curves cross faces at most one time

Resolving a Single Face with One Curve



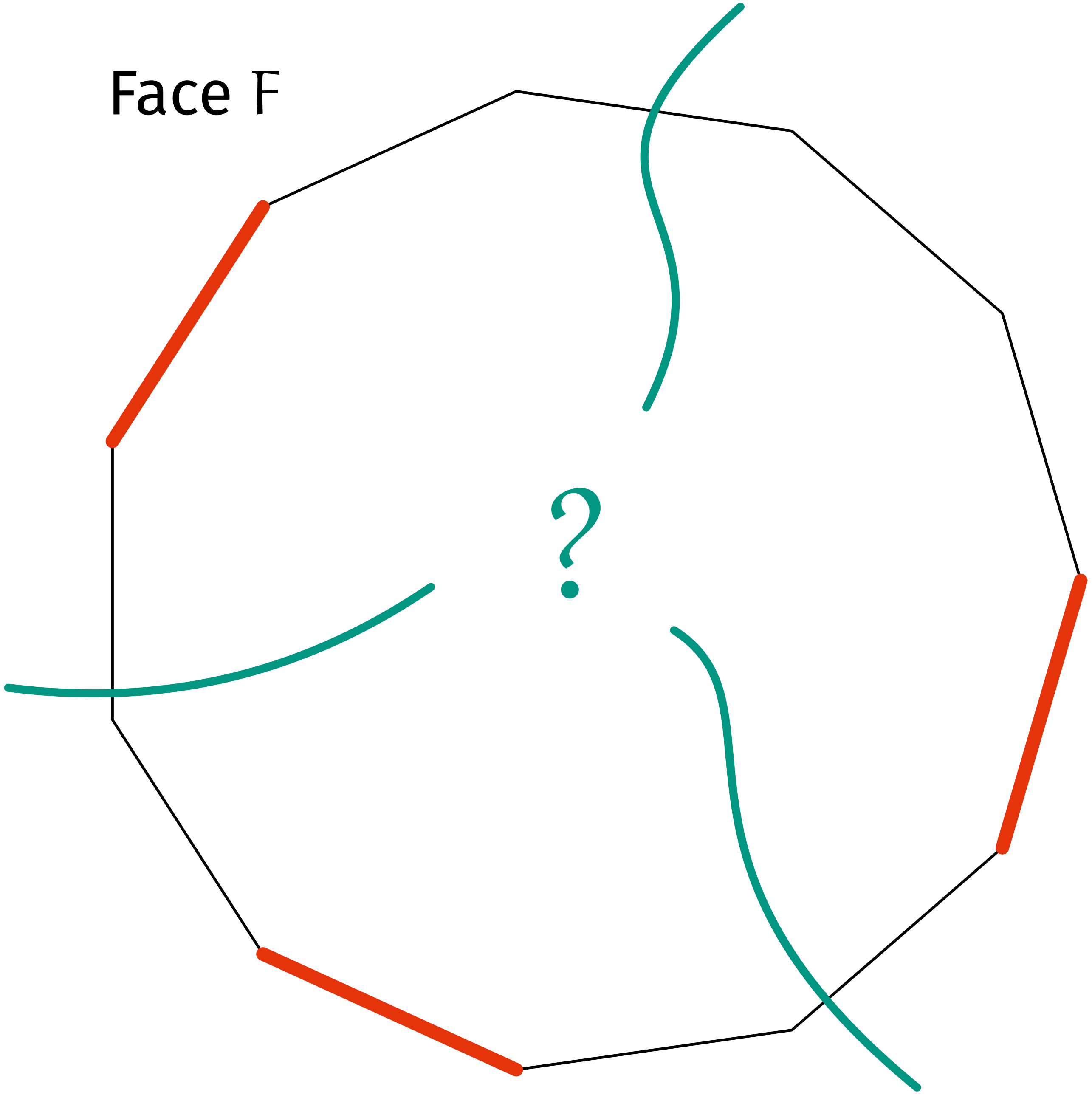
Resolution curves cross faces at most one time

Resolving a Single Face with One Curve



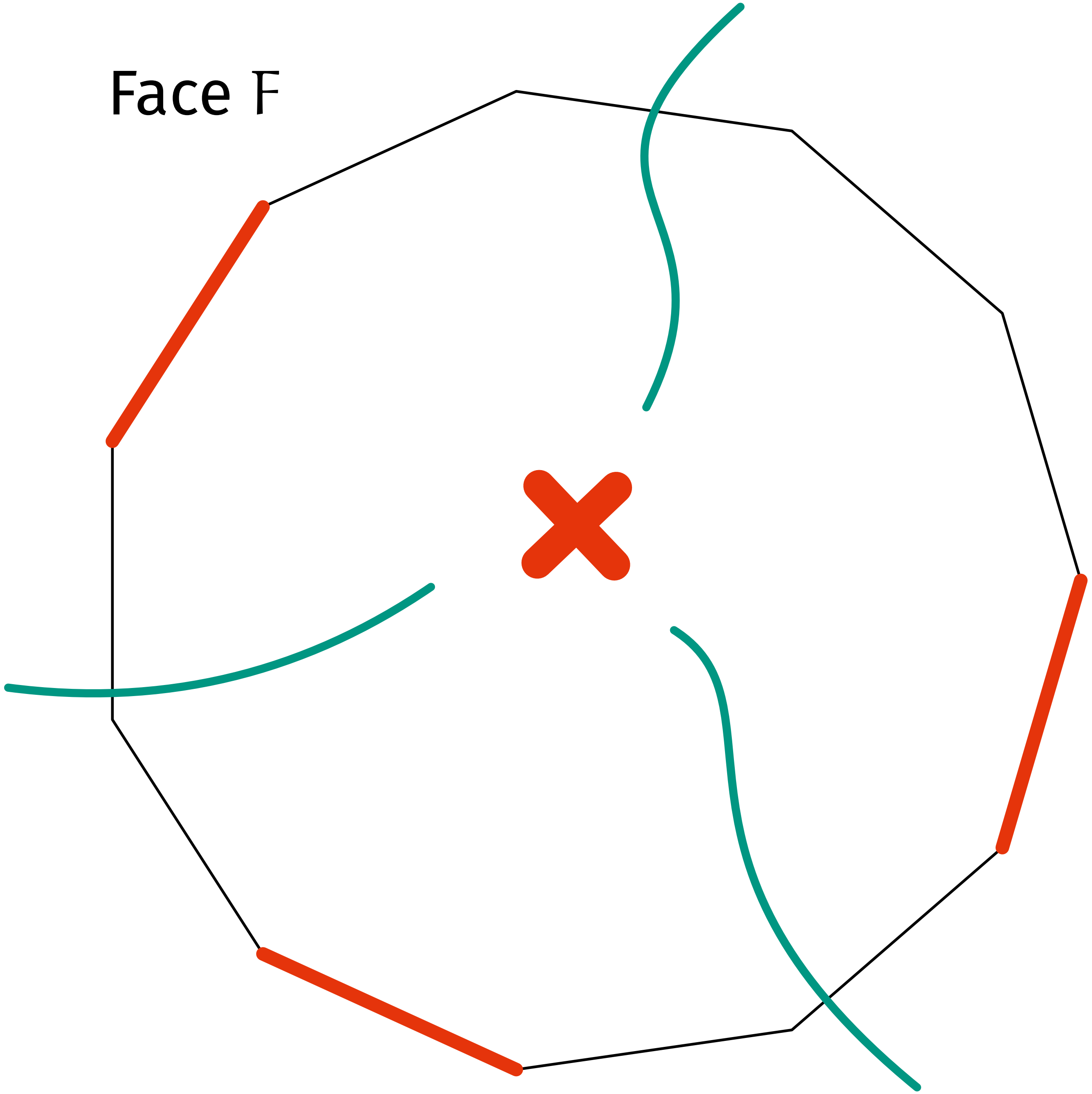
Resolution curves cross faces at most one time

Resolving a Single Face with One Curve



Resolution curves cross faces at most one time

Resolving a Single Face with One Curve

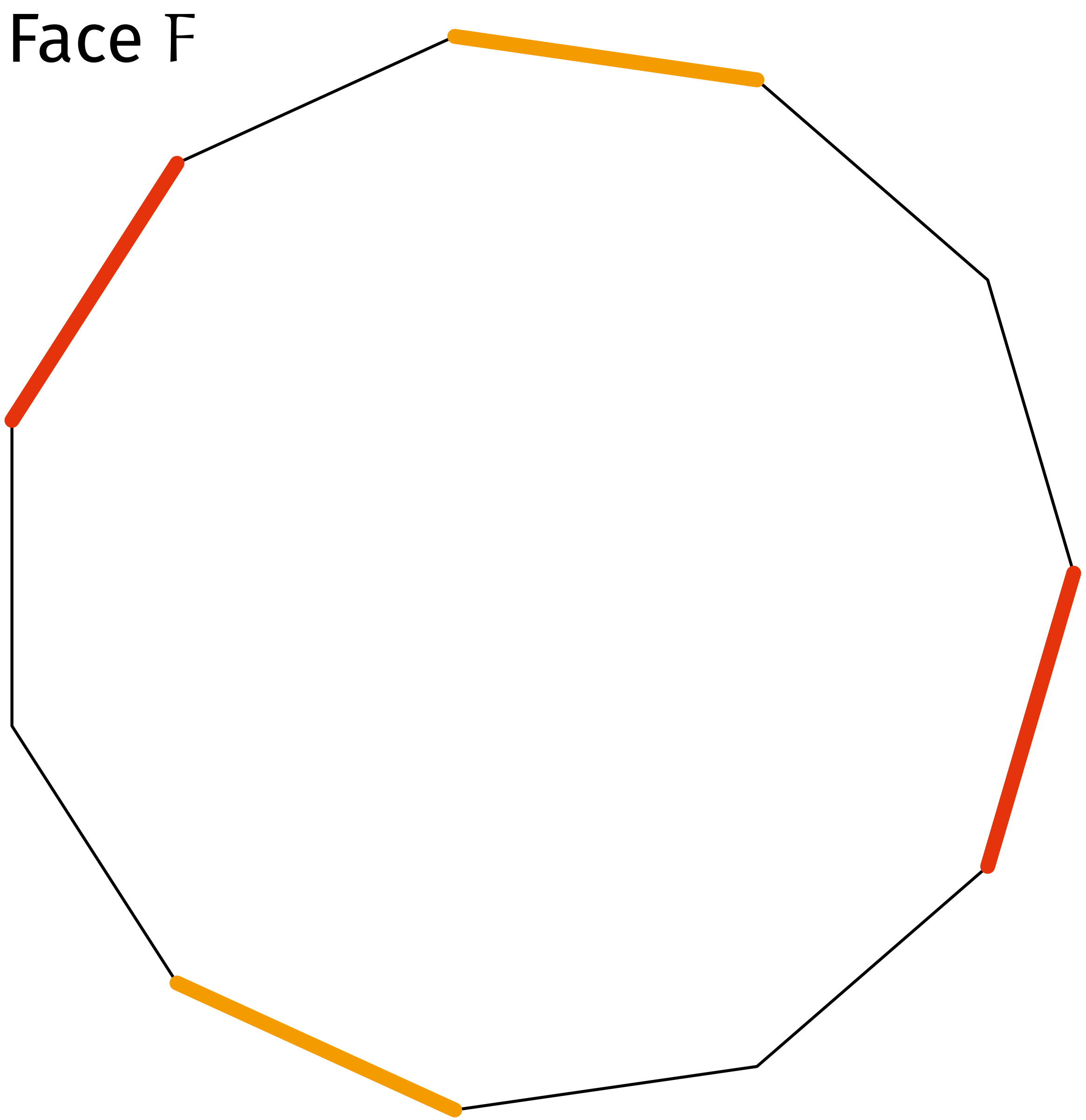


Resolution curves cross faces at most one time

Curves appear ≤ 2 times on the boundary of F

$\implies |F| \in \mathcal{O}(n)$

Resolving a Single Face with One Curve

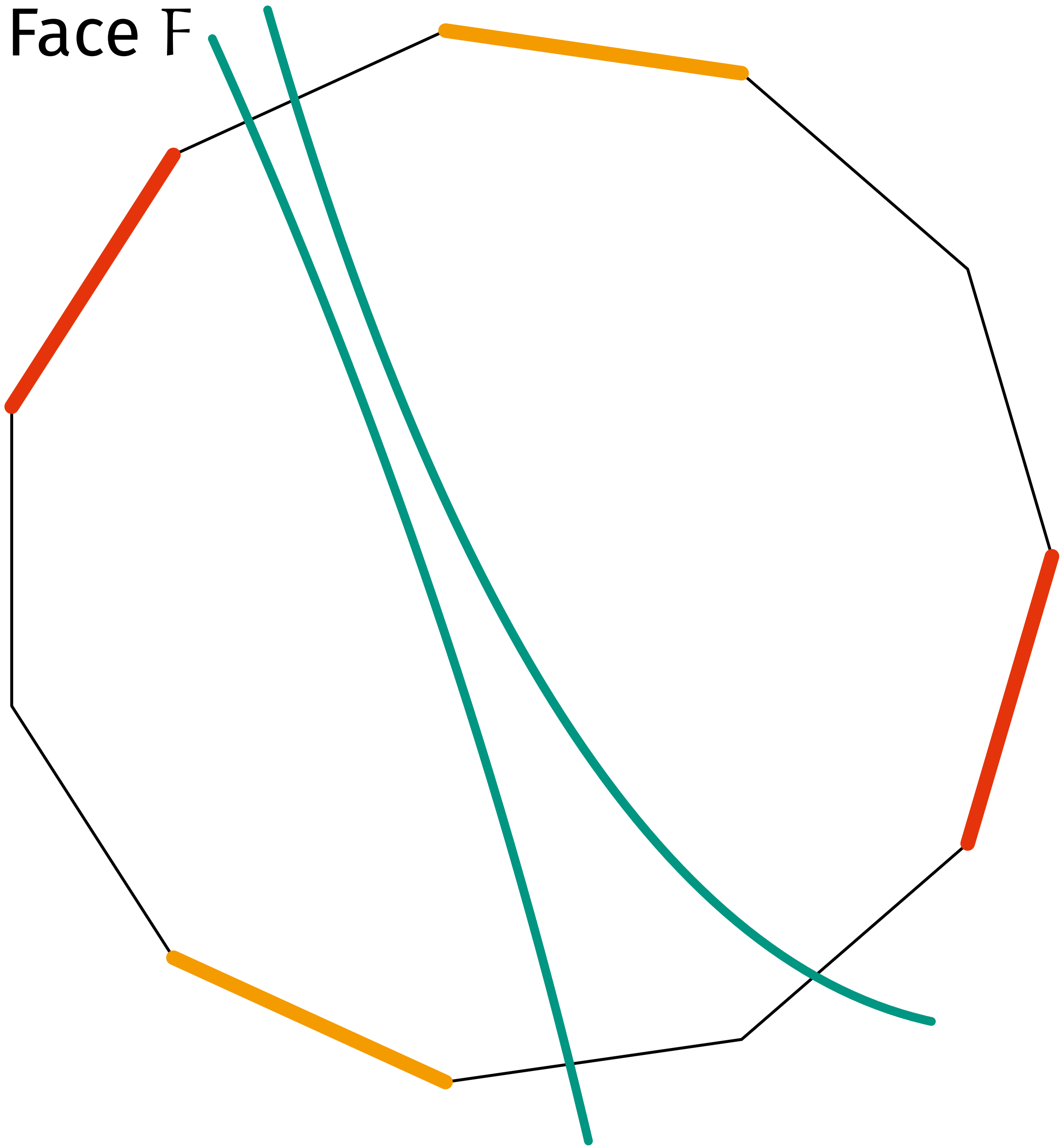


Resolution curves cross faces at most one time

Curves appear ≤ 2 times on the boundary of F

$$\implies |F| \in \mathcal{O}(n)$$

Resolving a Single Face with One Curve

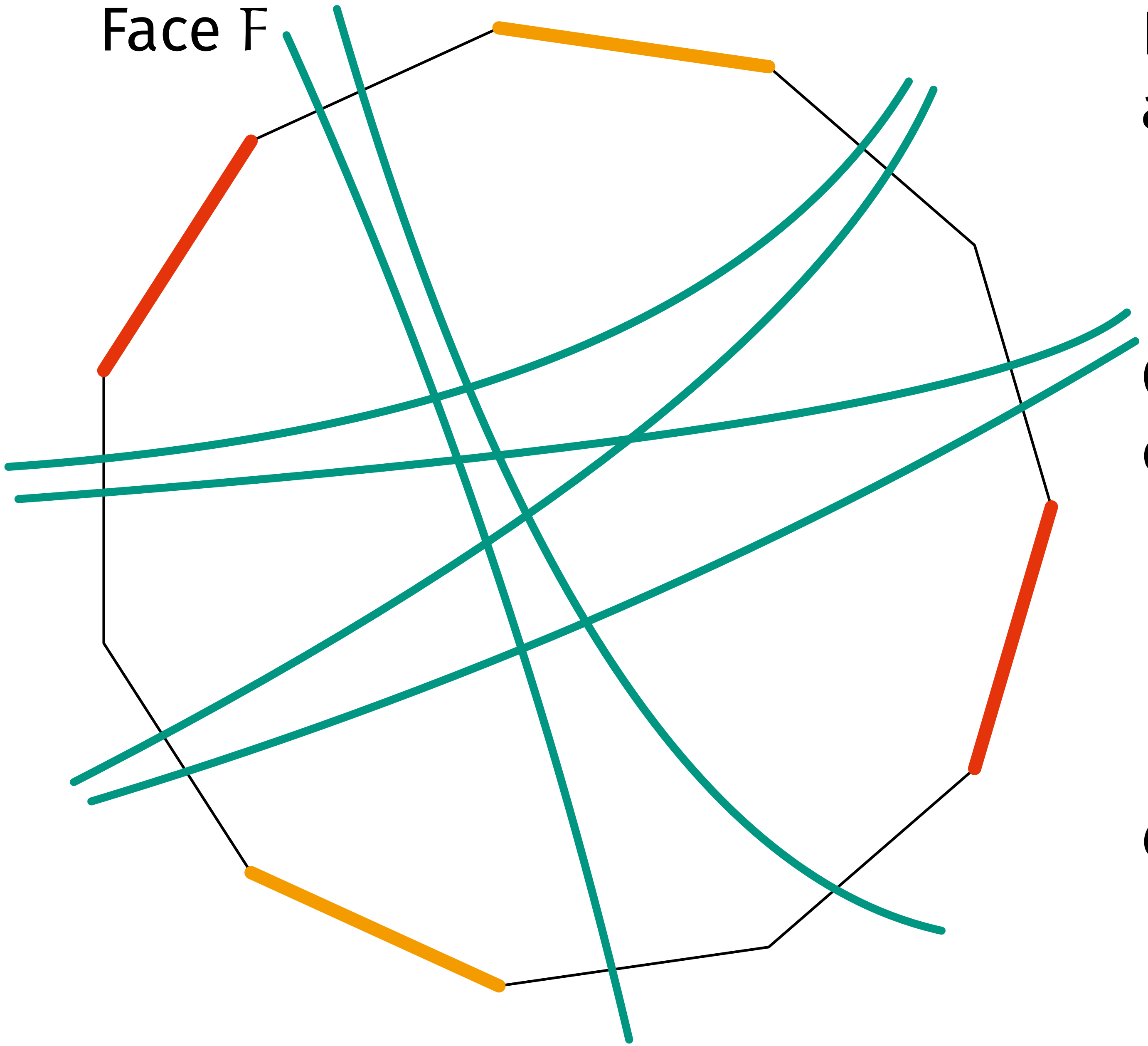


Resolution curves cross faces at most one time

Curves appear ≤ 2 times on the boundary of F

$\implies |F| \in \mathcal{O}(n)$

Resolving a Single Face with One Curve



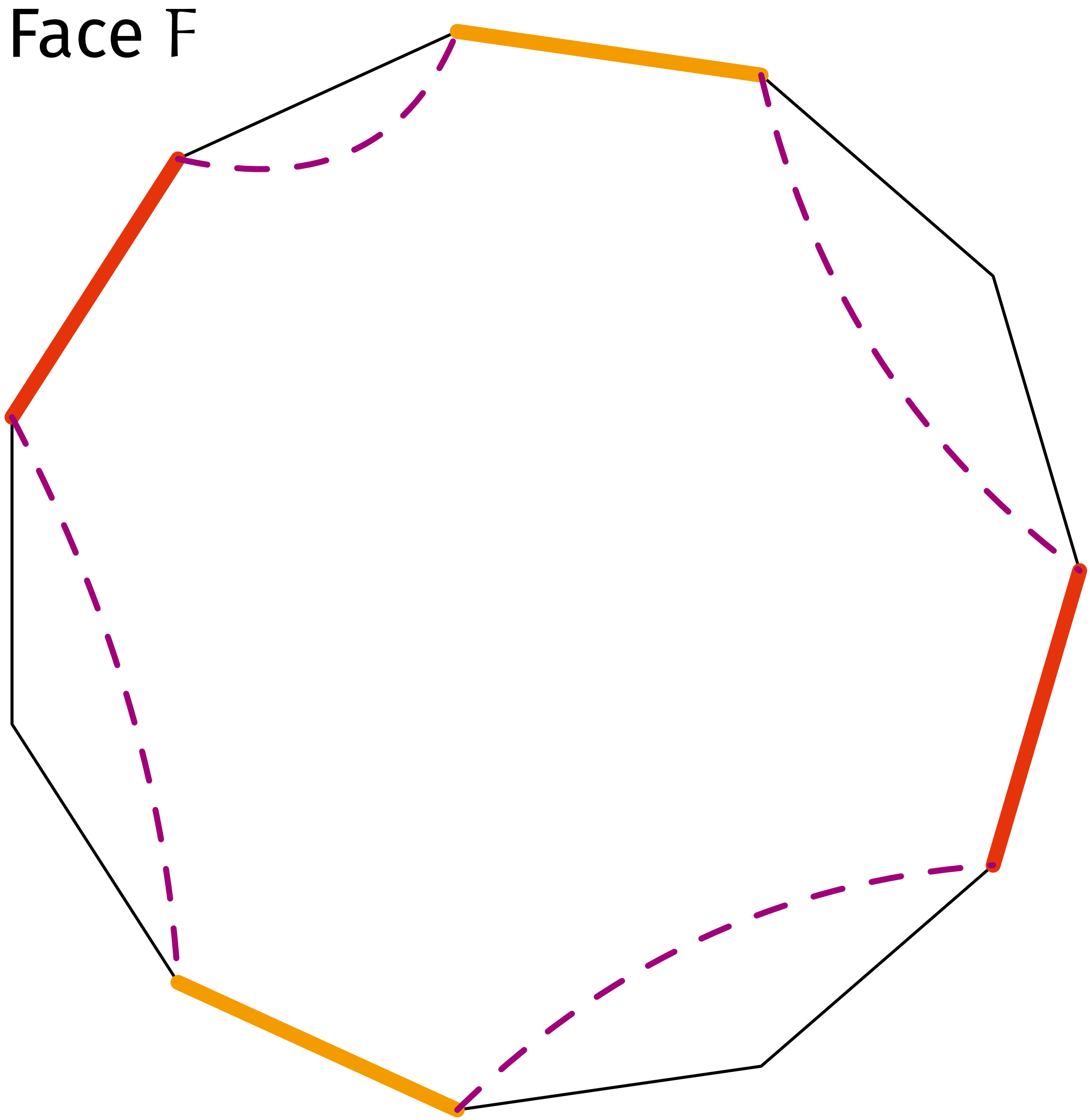
Resolution curves cross faces at most one time

Curves appear ≤ 2 times on the boundary of F

$\implies |F| \in \mathcal{O}(n)$

$\mathcal{O}(n^2)$ possibilities to resolve F

Resolving a Single Face with One Curve



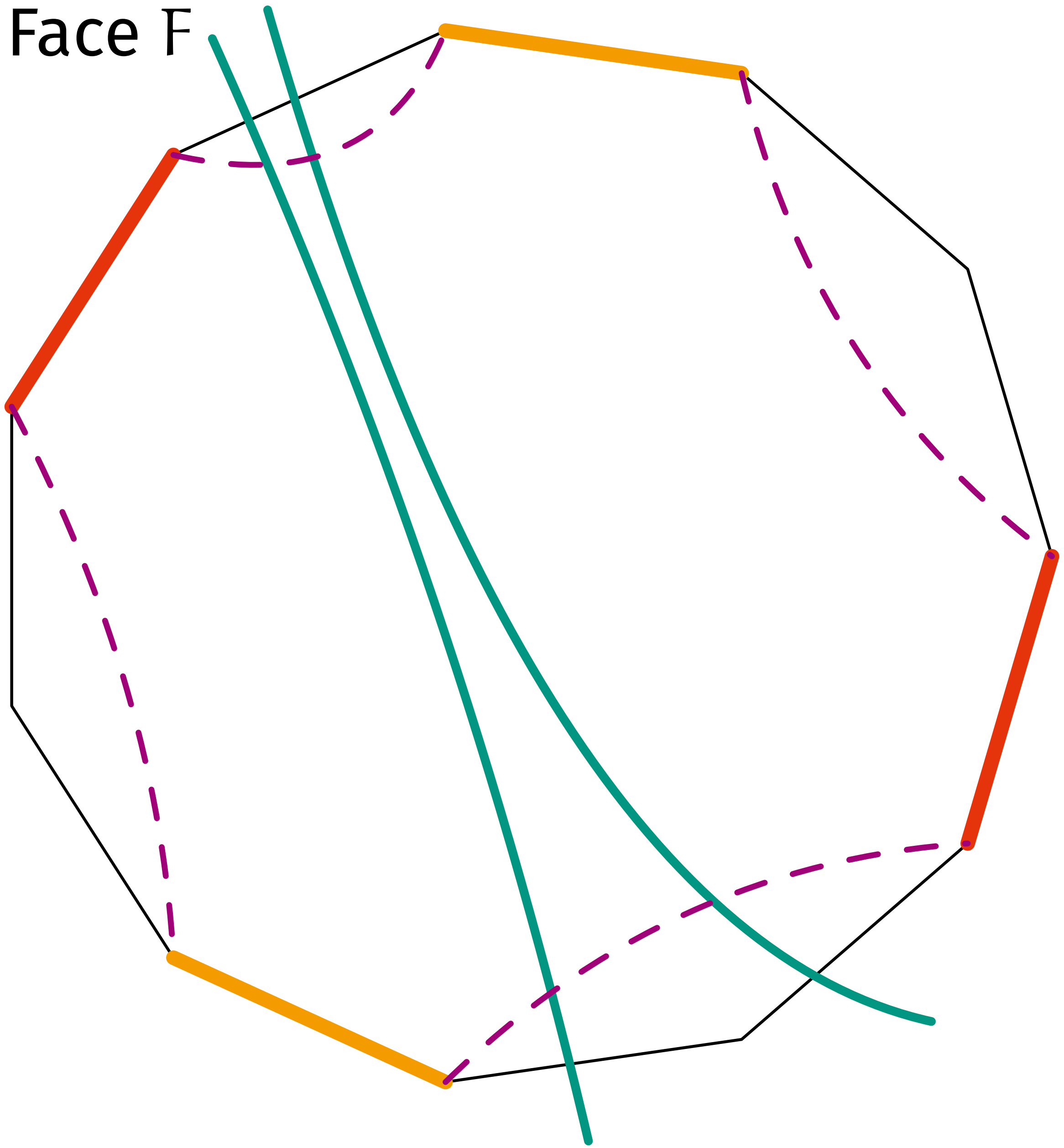
Resolution curves cross faces at most one time

Curves appear ≤ 2 times on the boundary of F

$\implies |F| \in \mathcal{O}(n)$

$\mathcal{O}(n^2)$ possibilities to resolve F

Resolving a Single Face with One Curve



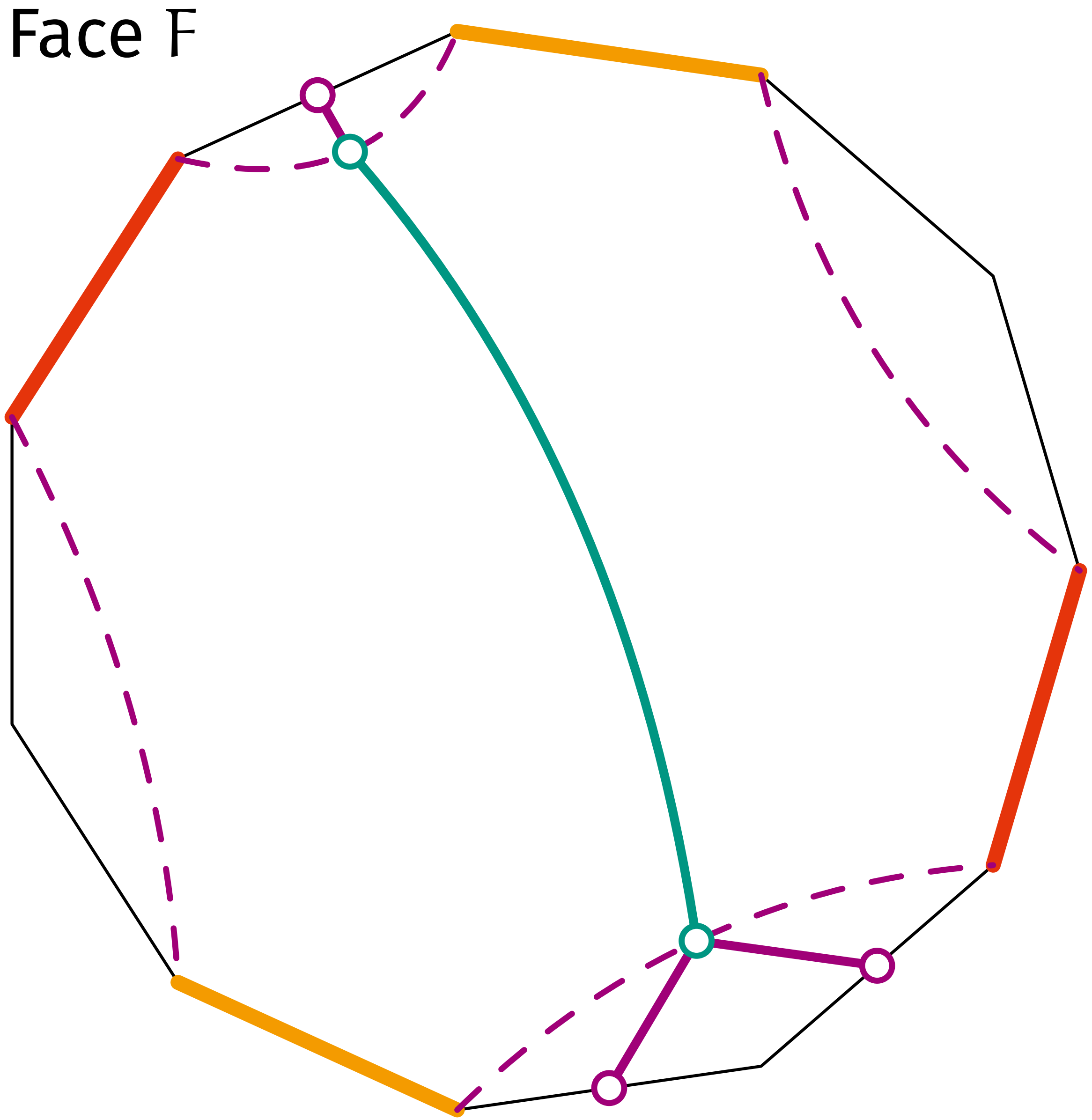
Resolution curves cross faces at most one time

Curves appear ≤ 2 times on the boundary of F

$\implies |F| \in \mathcal{O}(n)$

$\mathcal{O}(n^2)$ possibilities to resolve F

Resolving a Single Face with One Curve



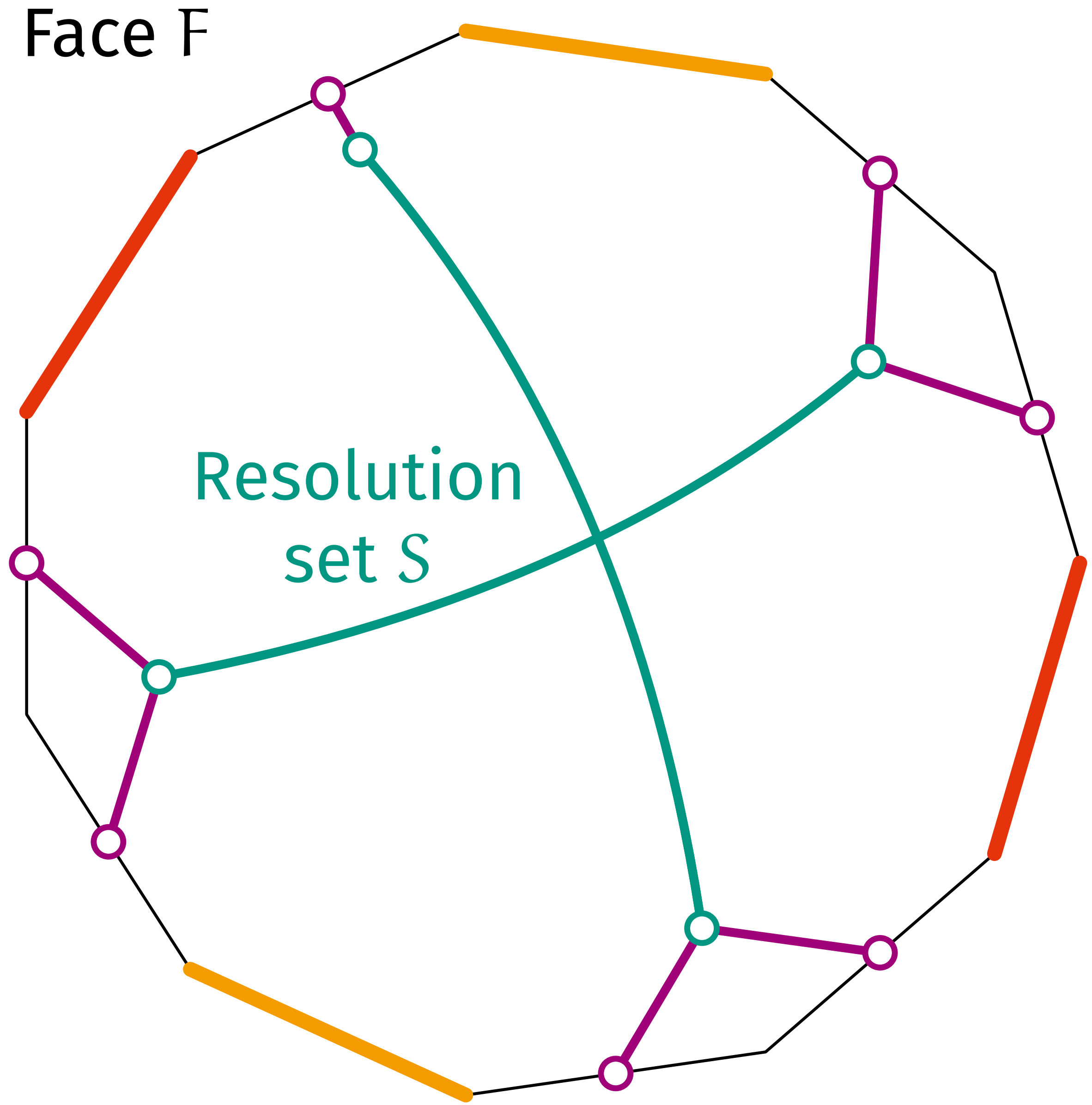
Resolution curves cross faces at most one time

Curves appear ≤ 2 times on the boundary of F

$\implies |F| \in \mathcal{O}(n)$

$\mathcal{O}(n^2)$ possibilities to resolve F

Resolving a Single Face with One Curve



Resolution curves cross faces at most one time

Curves appear ≤ 2 times on the boundary of F

$\implies |F| \in \mathcal{O}(n)$

$\mathcal{O}(n^2)$ possibilities to resolve F

Outline

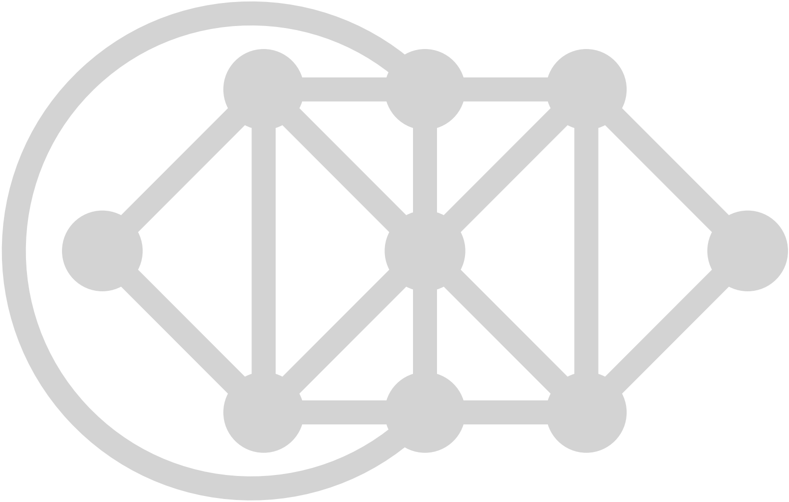
Nonograms

How to remove popular faces

Resolution with one curve is NP-complete...

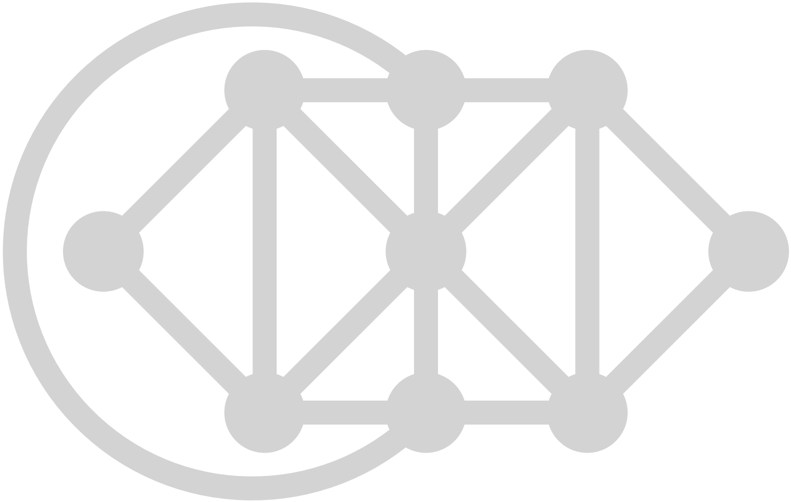
Adding a Single Curve is NP-complete

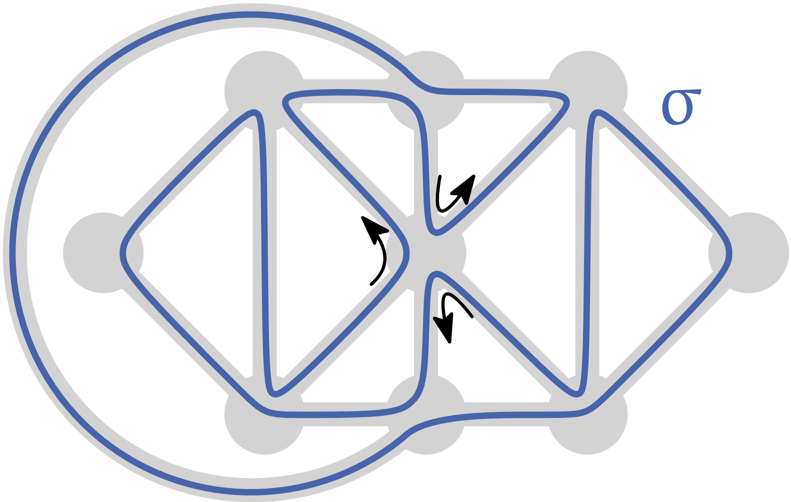
Given: Graph G, embedded in $\mathbb{R}^2$



Adding a Single Curve is NP-complete

Given: Graph G, embedded in $\mathbb{R}^2$



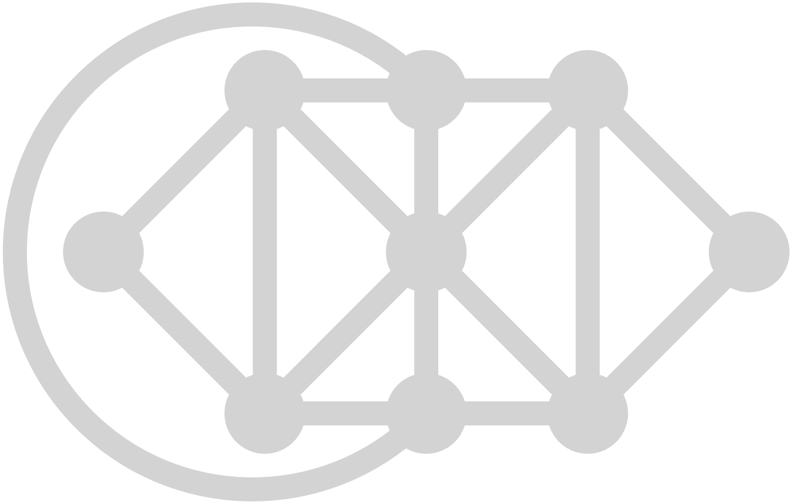


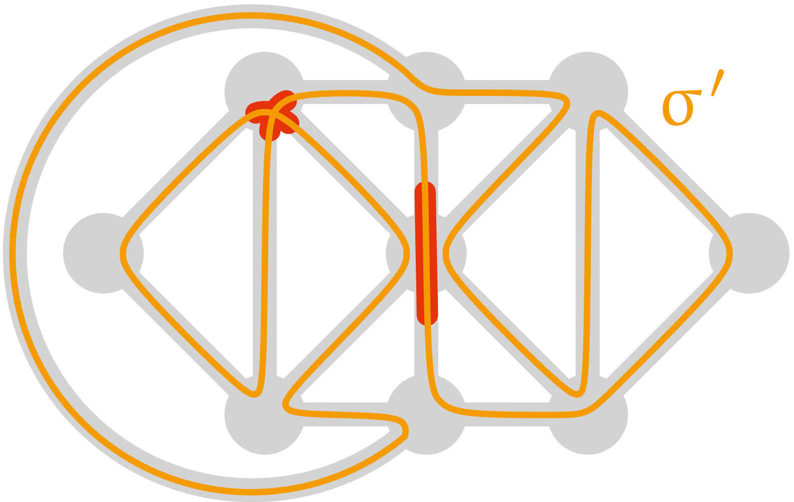
Non-Crossing Eulerian Cycle

[Bent & Manber, ’87]

Adding a Single Curve is NP-complete

Given: Graph G, embedded in $\mathbb{R}^2$



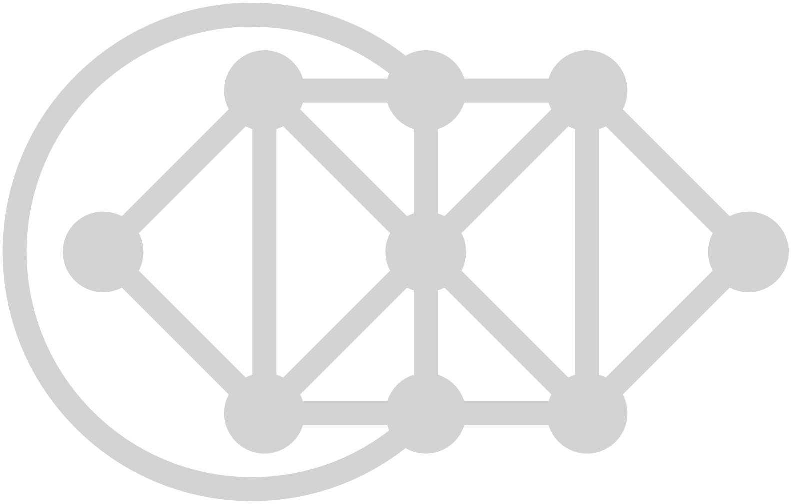


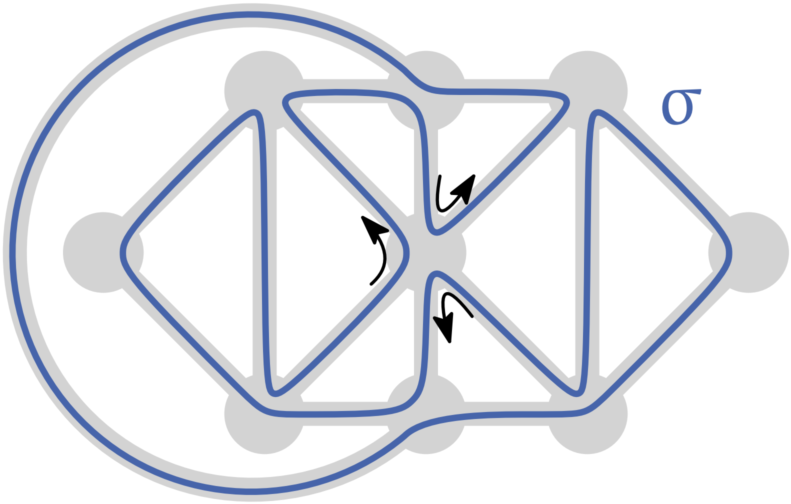
Non-Crossing Eulerian Cycle

[Bent & Manber, '87]

Adding a Single Curve is NP-complete

Given: Graph G, embedded in $\mathbb{R}^2$



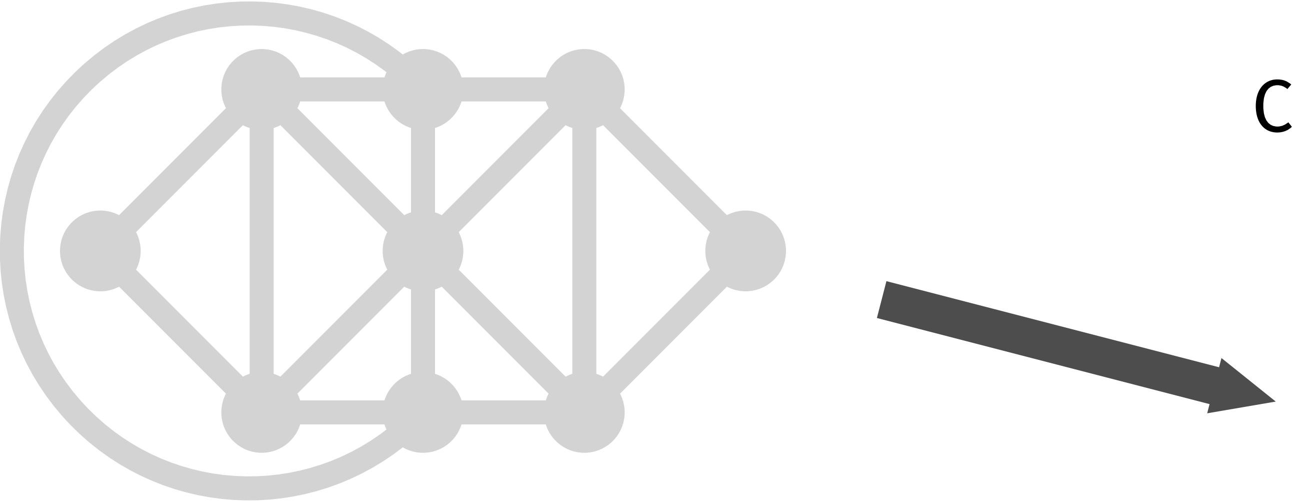


Non-Crossing Eulerian Cycle

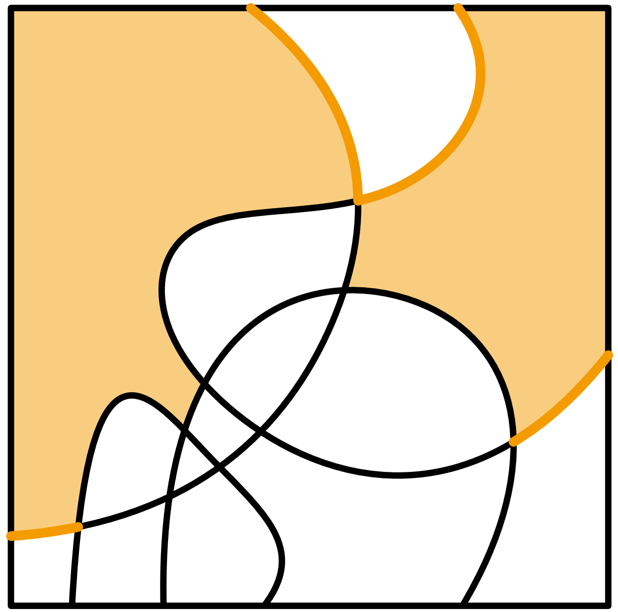
[Bent & Manber, '87]

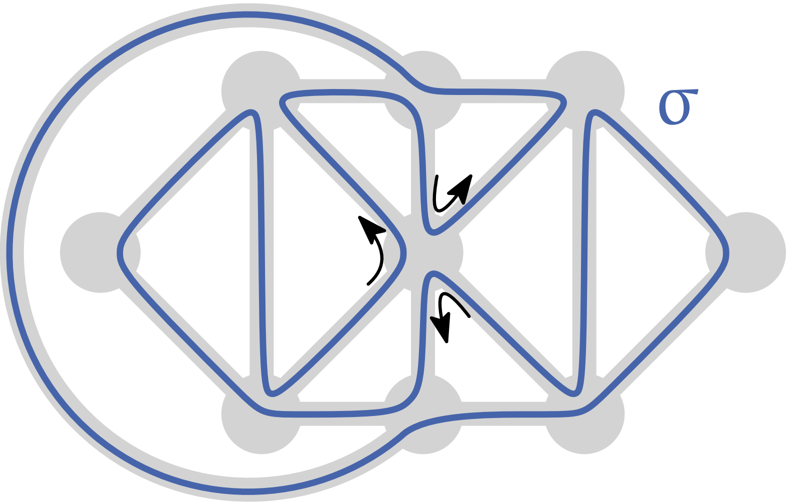
Adding a Single Curve is NP-complete

Given: Graph G, embedded in $\mathbb{R}^2$



Curve arrangement $\mathcal{A}$ (advanced)



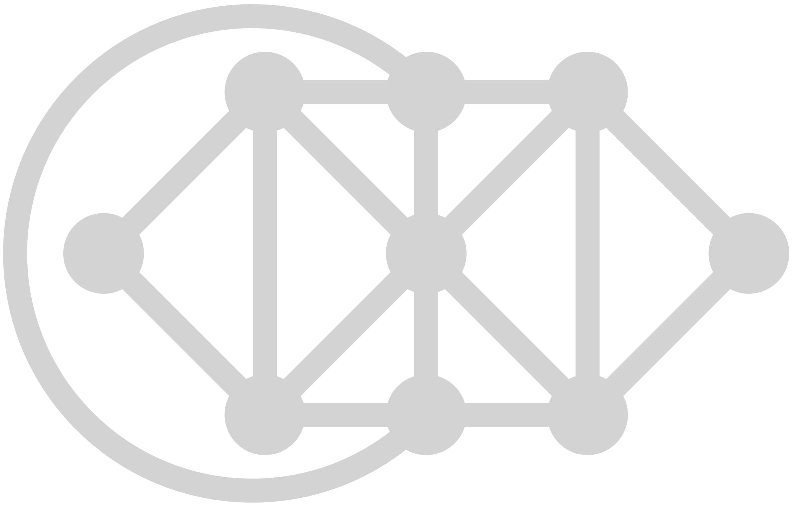


Non-Crossing Eulerian Cycle

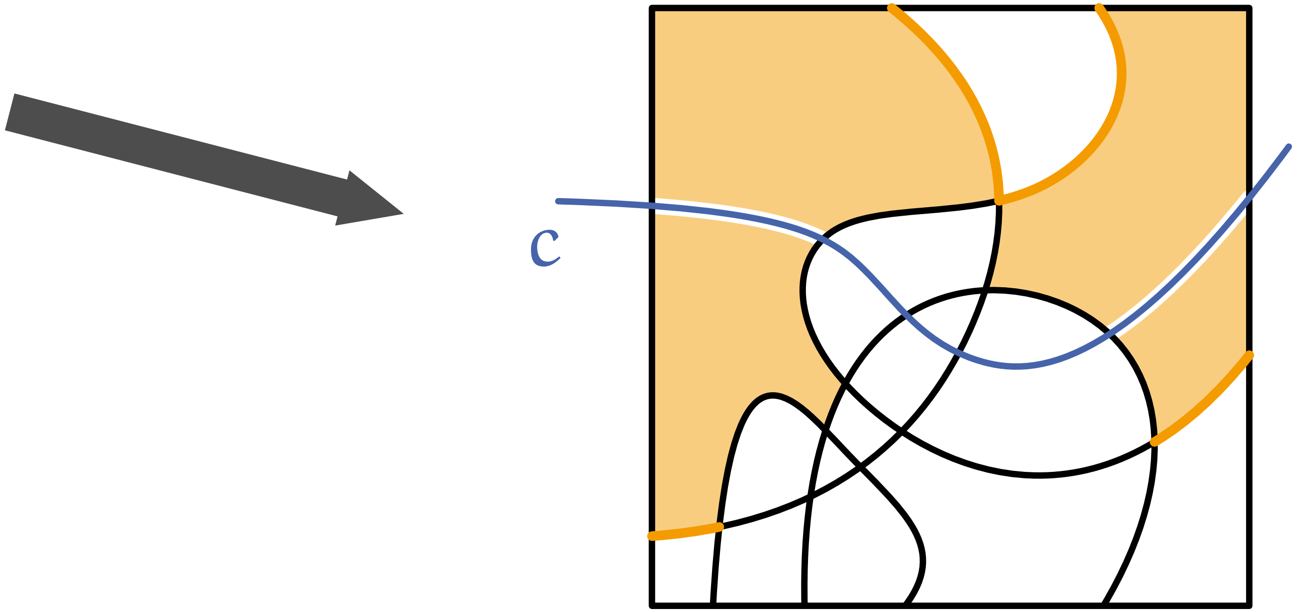
[Bent & Manber, '87]

Adding a Single Curve is NP-complete

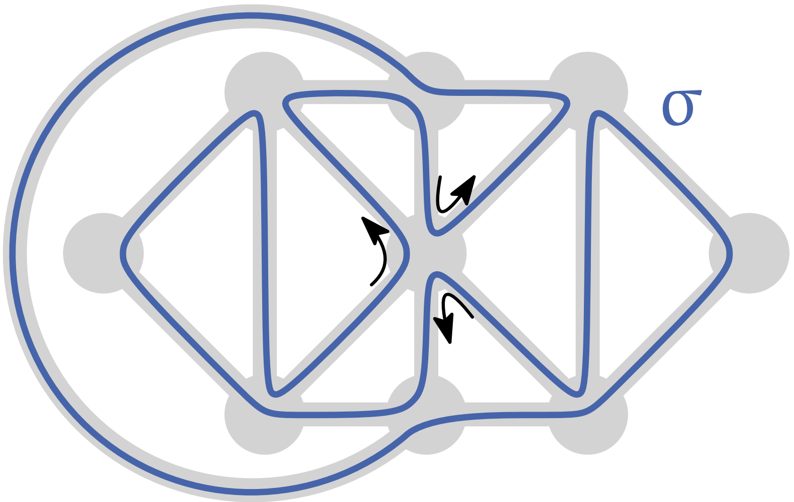
Given: Graph G, embedded in $\mathbb{R}^2$



Curve arrangement $\mathcal{A}$ (advanced)



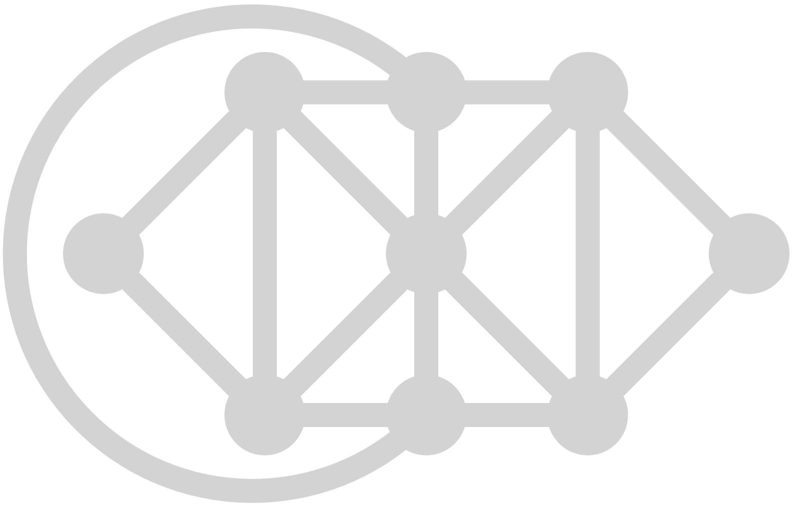
Can be made basic
with 1 resolution curve c



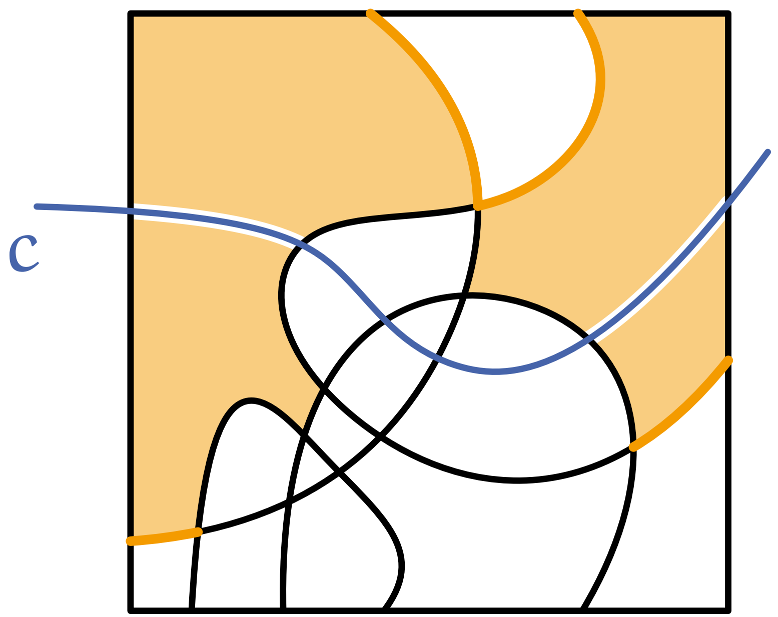
Non-Crossing Eulerian Cycle
[Bent & Manber, ’87]

Adding a Single Curve is NP-complete

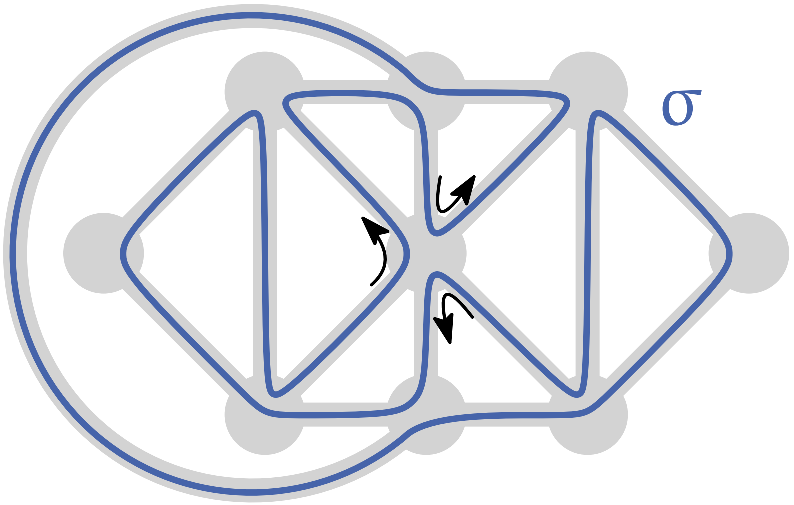
Given: Graph G, embedded in $\mathbb{R}^2$



Curve arrangement $\mathcal{A}$ (advanced)



Can be made basic
with 1 resolution curve c



Non-Crossing Eulerian Cycle

[Bent & Manber, '87]

Outline

Nonograms

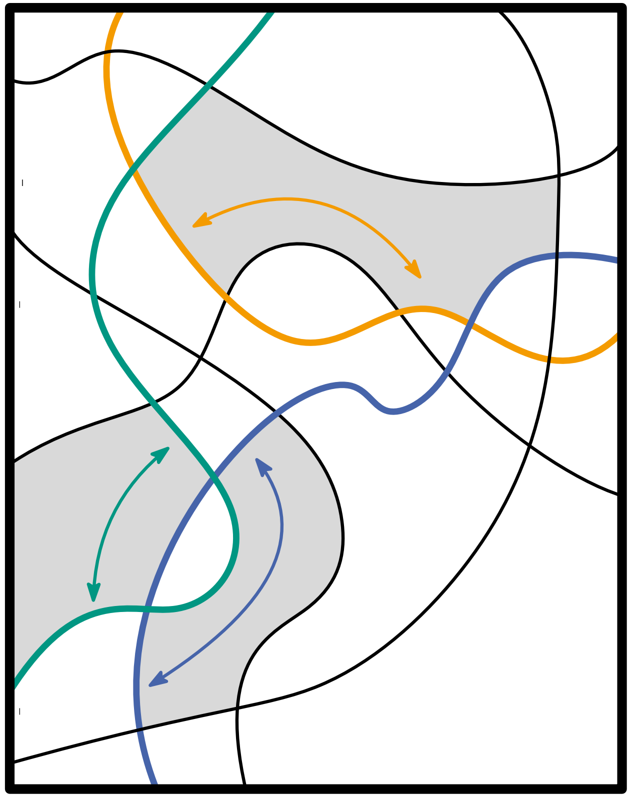
How to remove popular faces

Resolution with one curve is NP-complete...

...but we can do it randomized in FPT time

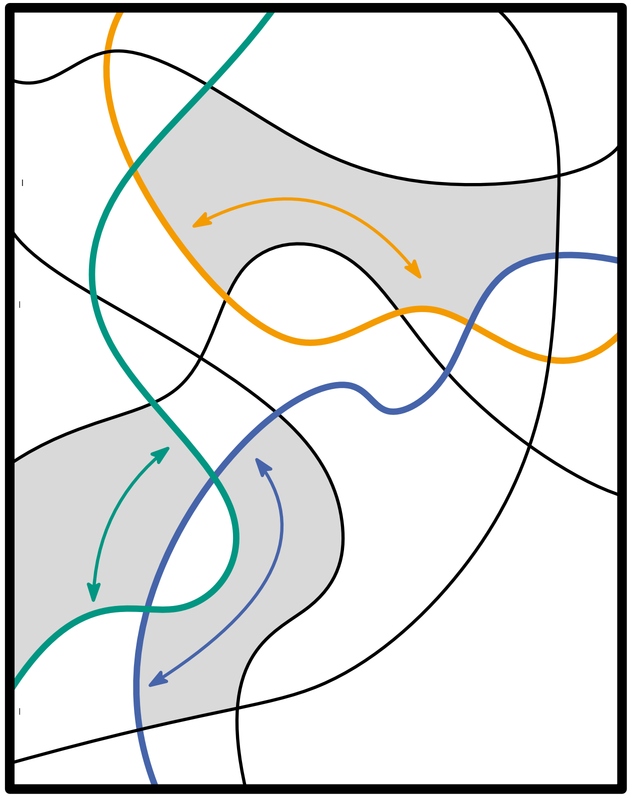
Representing a Resolution Curve

Curve arrangement $\mathcal{A}$



Representing a Resolution Curve

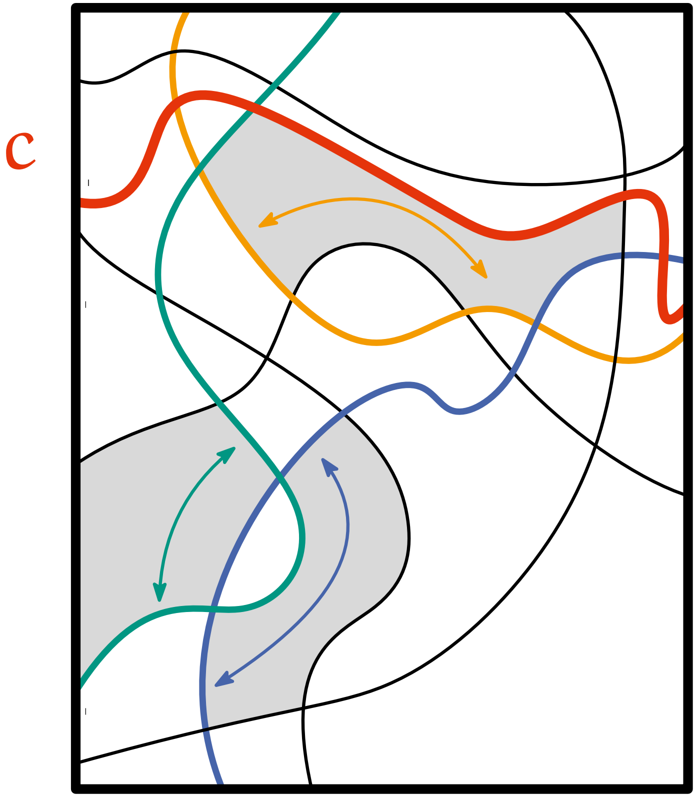
Curve arrangement $\mathcal{A}$



Representing a Resolution Curve

Curve arrangement $\mathcal{A}$

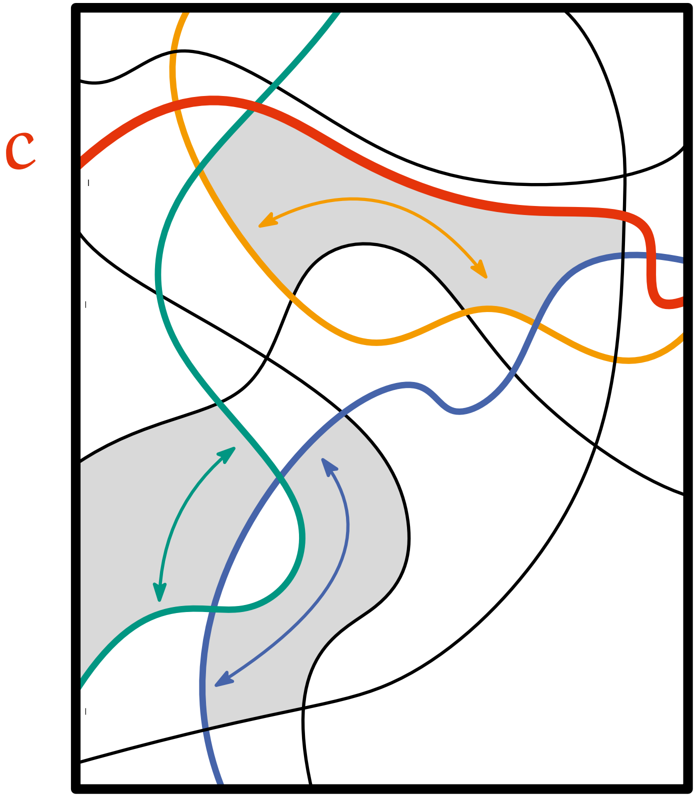
Additional curve c



Representing a Resolution Curve

Curve arrangement $\mathcal{A}$

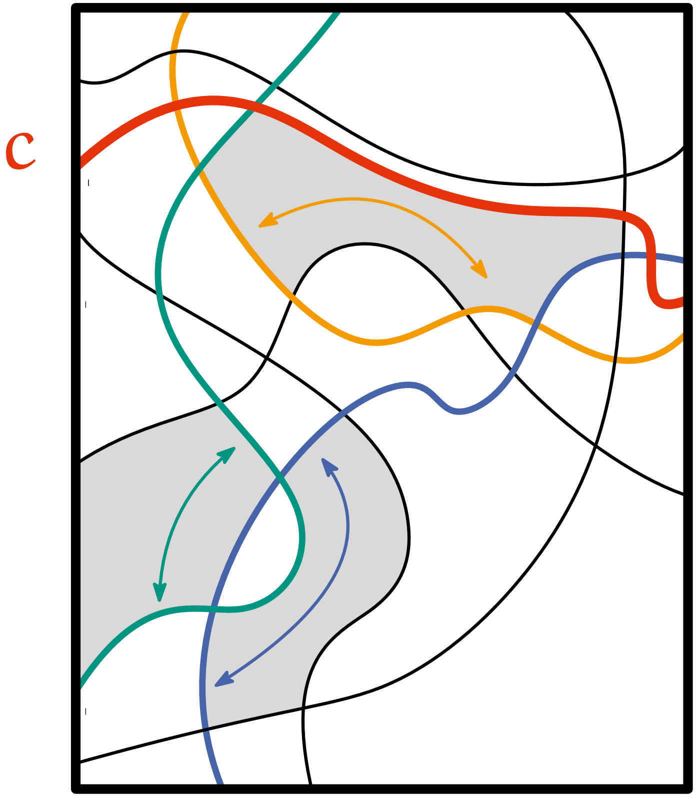
Additional curve c



Representing a Resolution Curve

Curve arrangement $\mathcal{A}$

Additional curve c

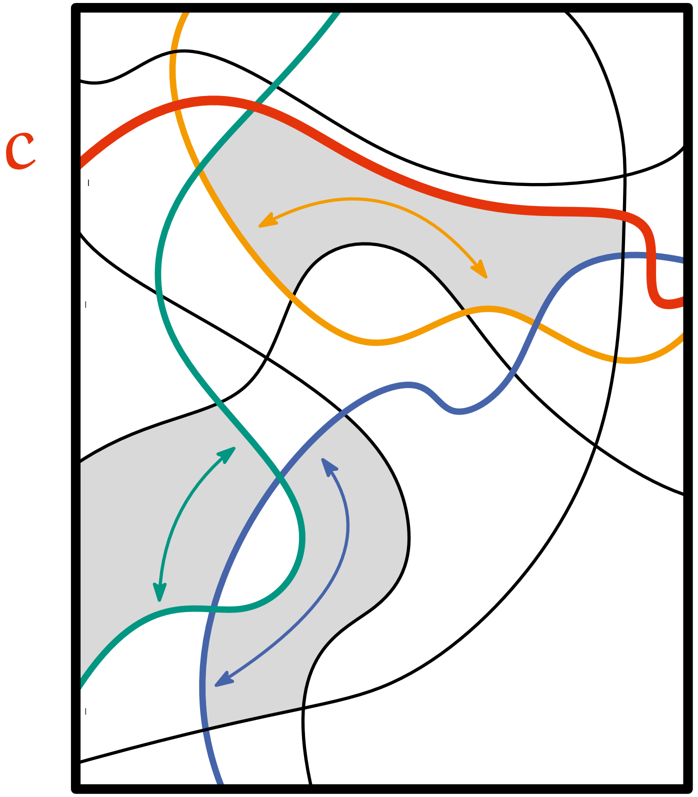


Dual graph $\mathcal{A}^{\mathrm{d}}$

Representing a Resolution Curve

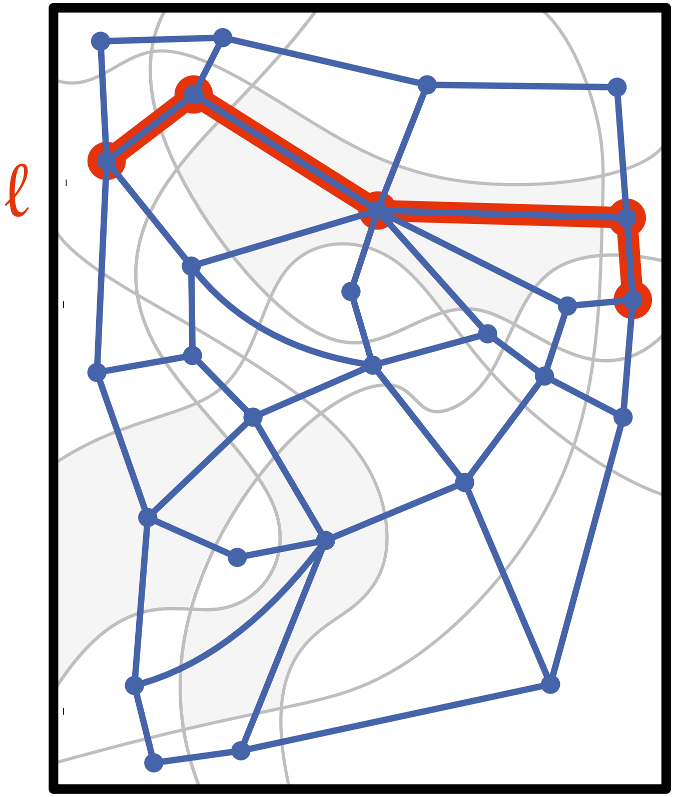
Curve arrangement $\mathcal{A}$

Additional curve c



Dual graph $\mathcal{A}^{\mathrm{d}}$

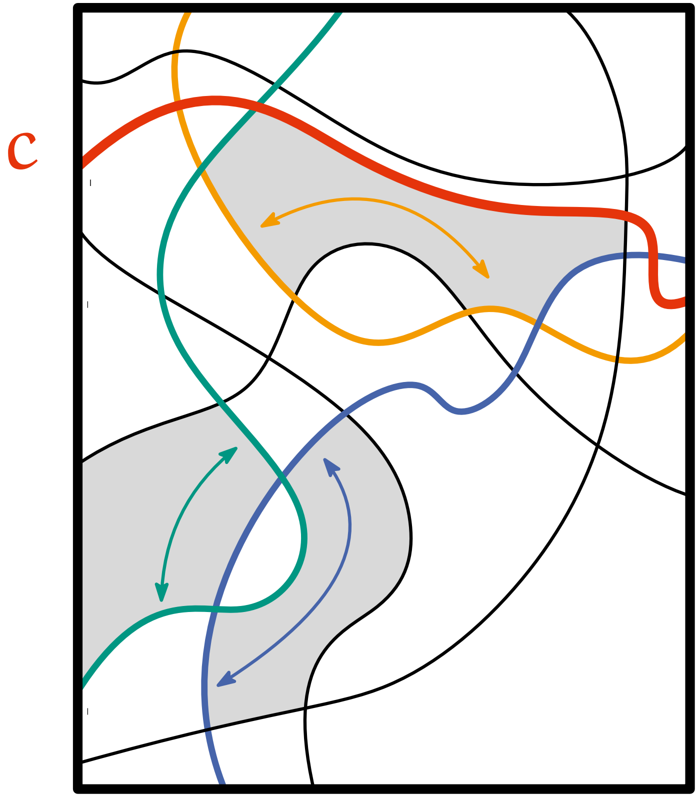
walk ℓ



Representing a Resolution Curve

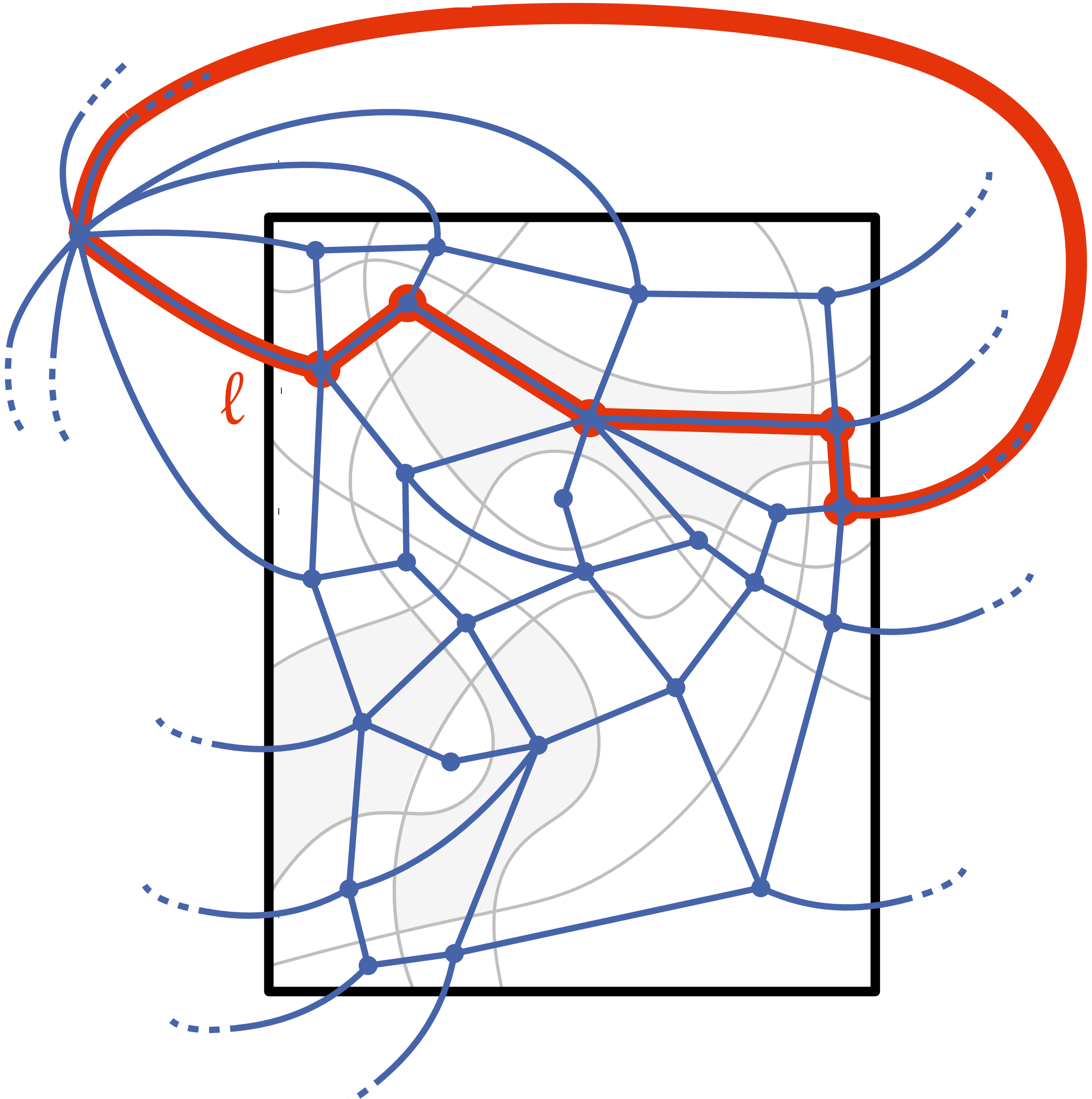
Curve arrangement $\mathcal{A}$

Additional curve c



Dual graph $\mathcal{A}^{\mathrm{d}}$

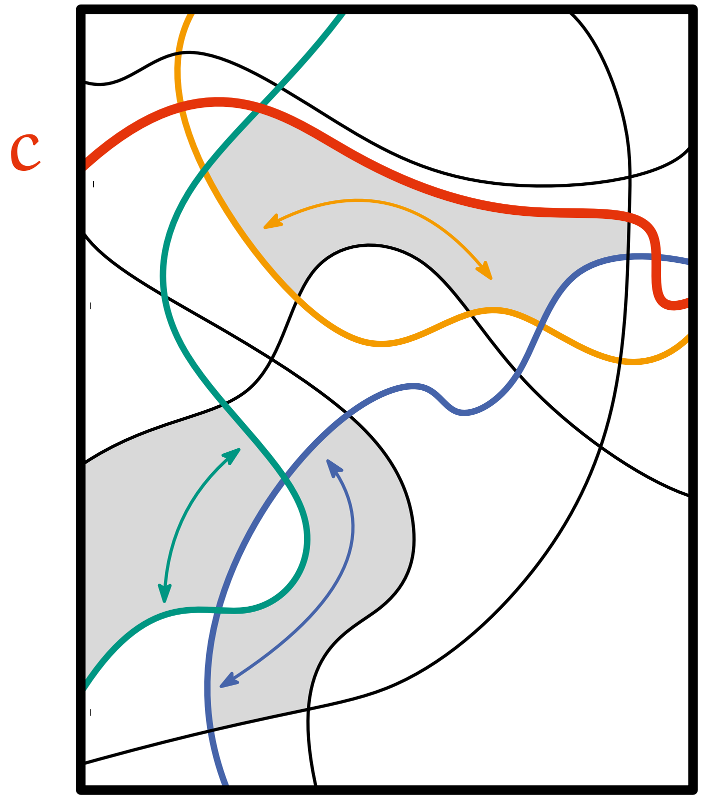
closed walk ℓ



Representing a Resolution Curve

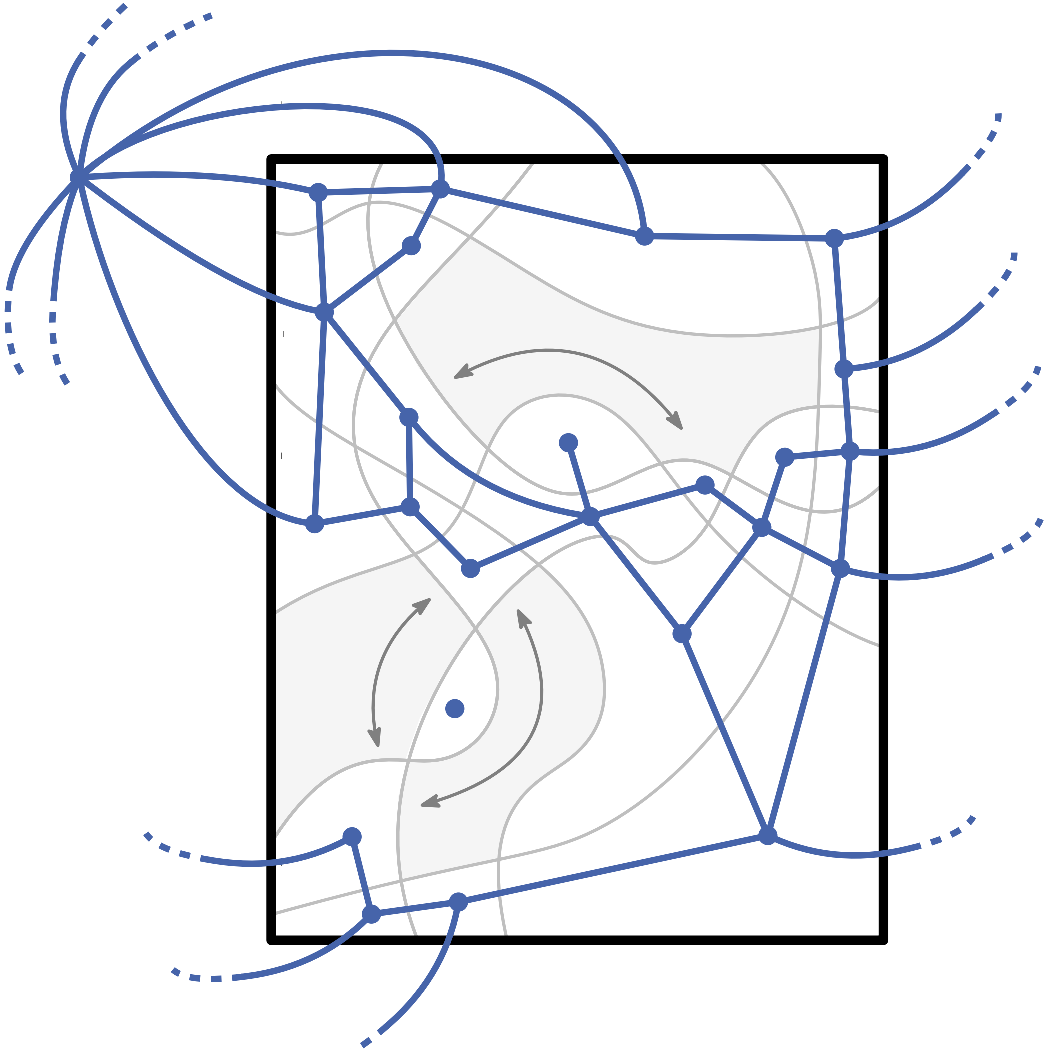
Curve arrangement $\mathcal{A}$

Additional curve c



Dual graph $\mathcal{A}^{\mathrm{d}}$

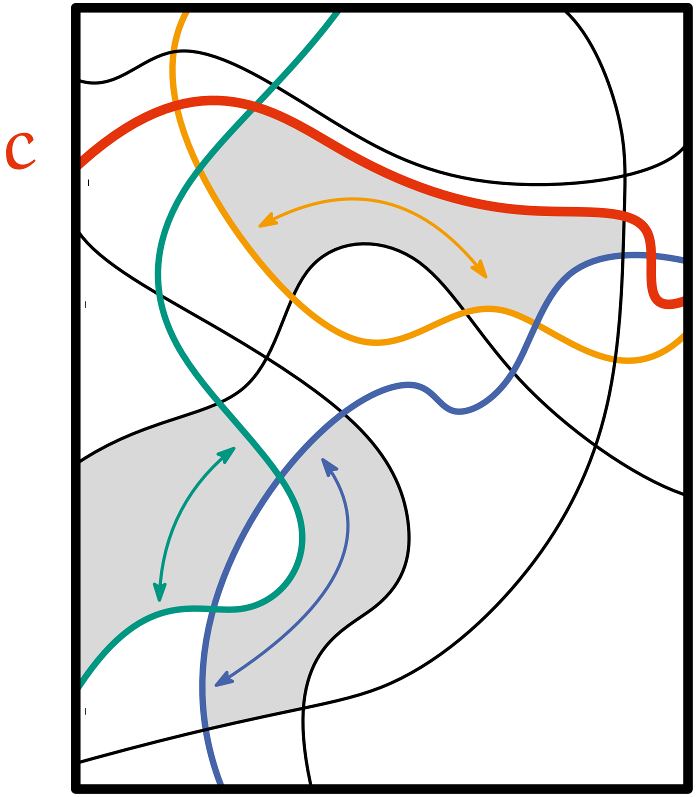
closed walk ℓ



Representing a Resolution Curve

Curve arrangement $\mathcal{A}$

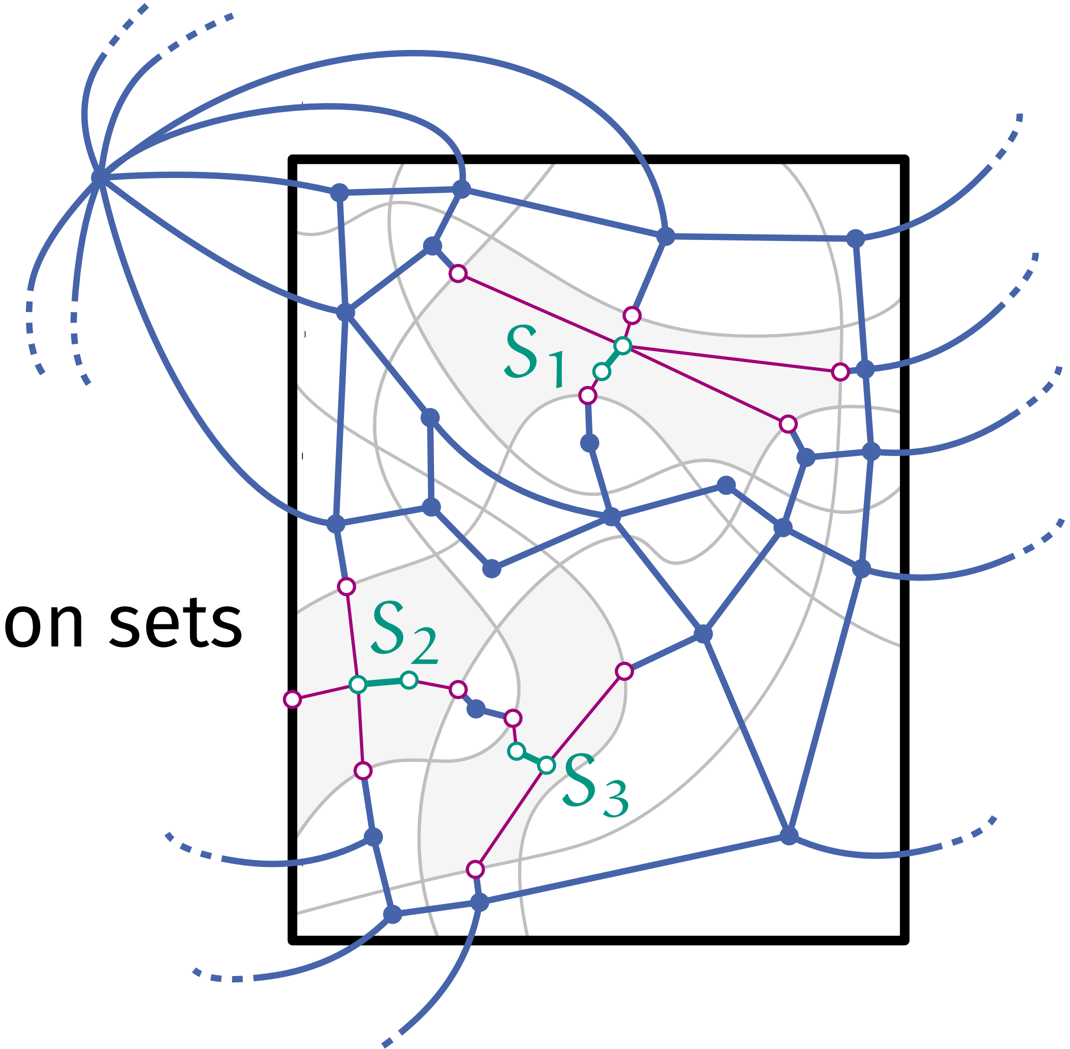
Additional curve c



Dual graph $\mathcal{A}^{\mathrm{d}}$

closed walk ℓ

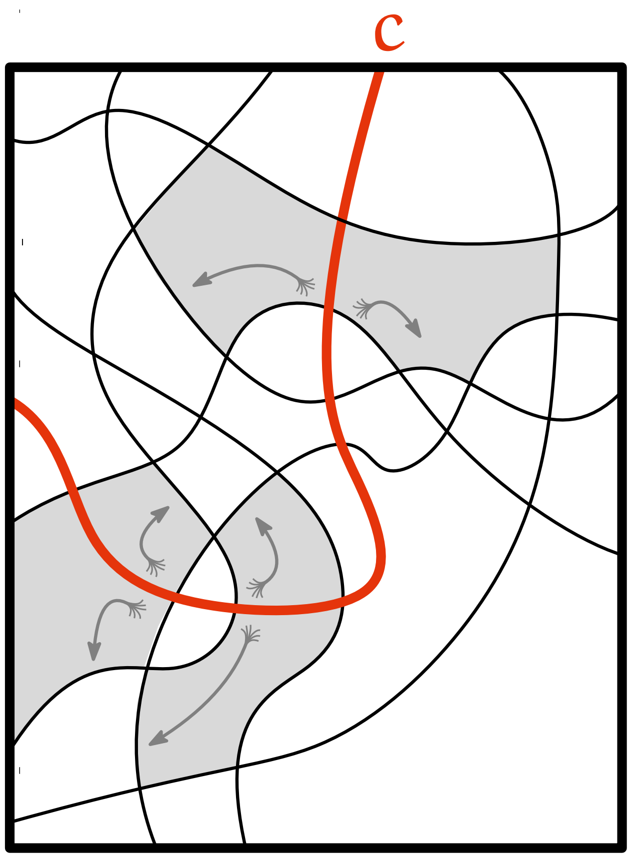
Resolution sets



Representing a Resolution Curve

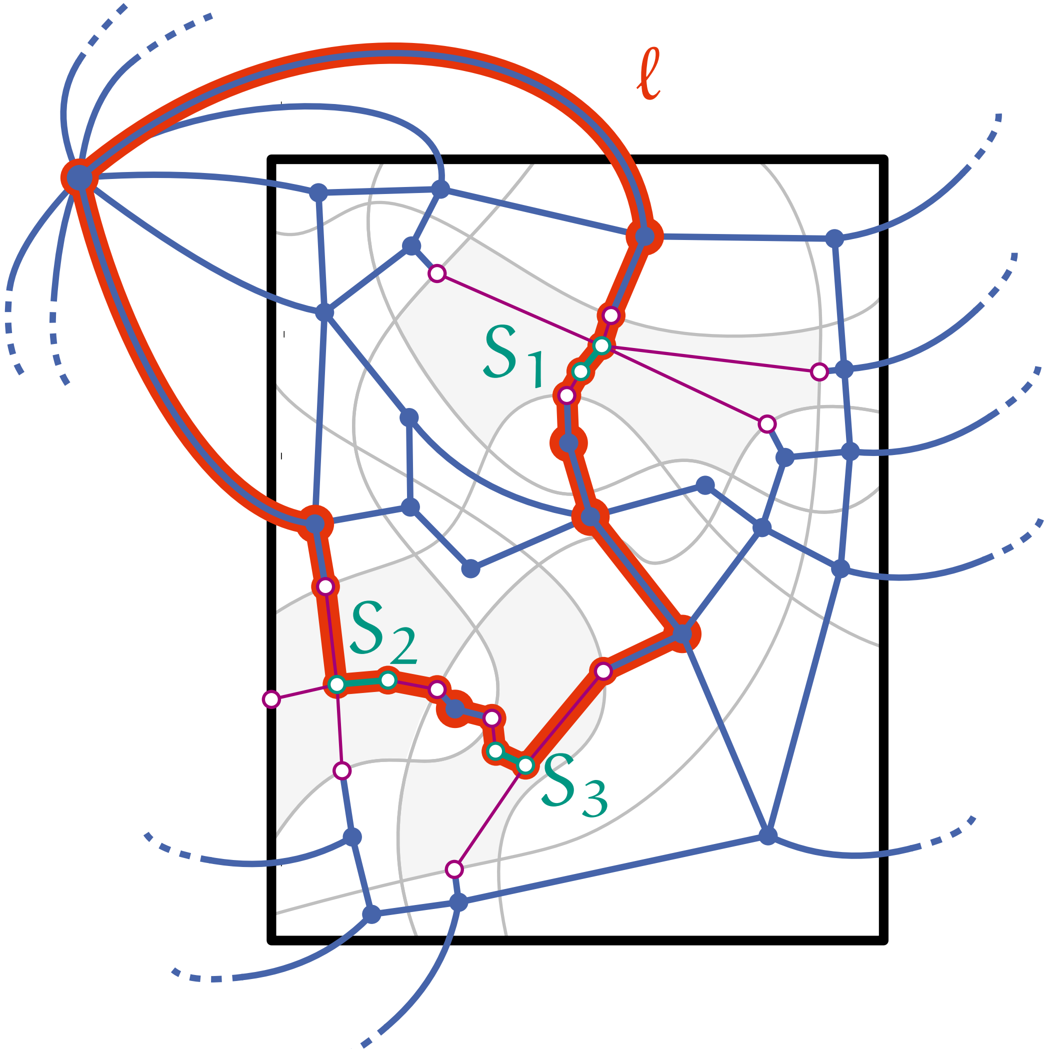
Curve arrangement $\mathcal{A}$

Additional curve c



Dual graph $\mathcal{A}^d$

closed walk ℓ



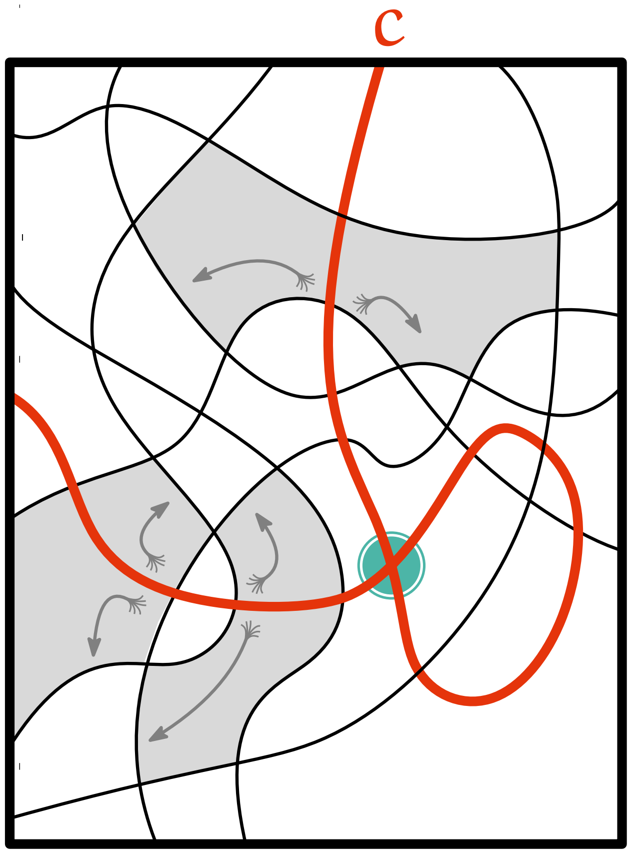
Representing a Resolution Curve

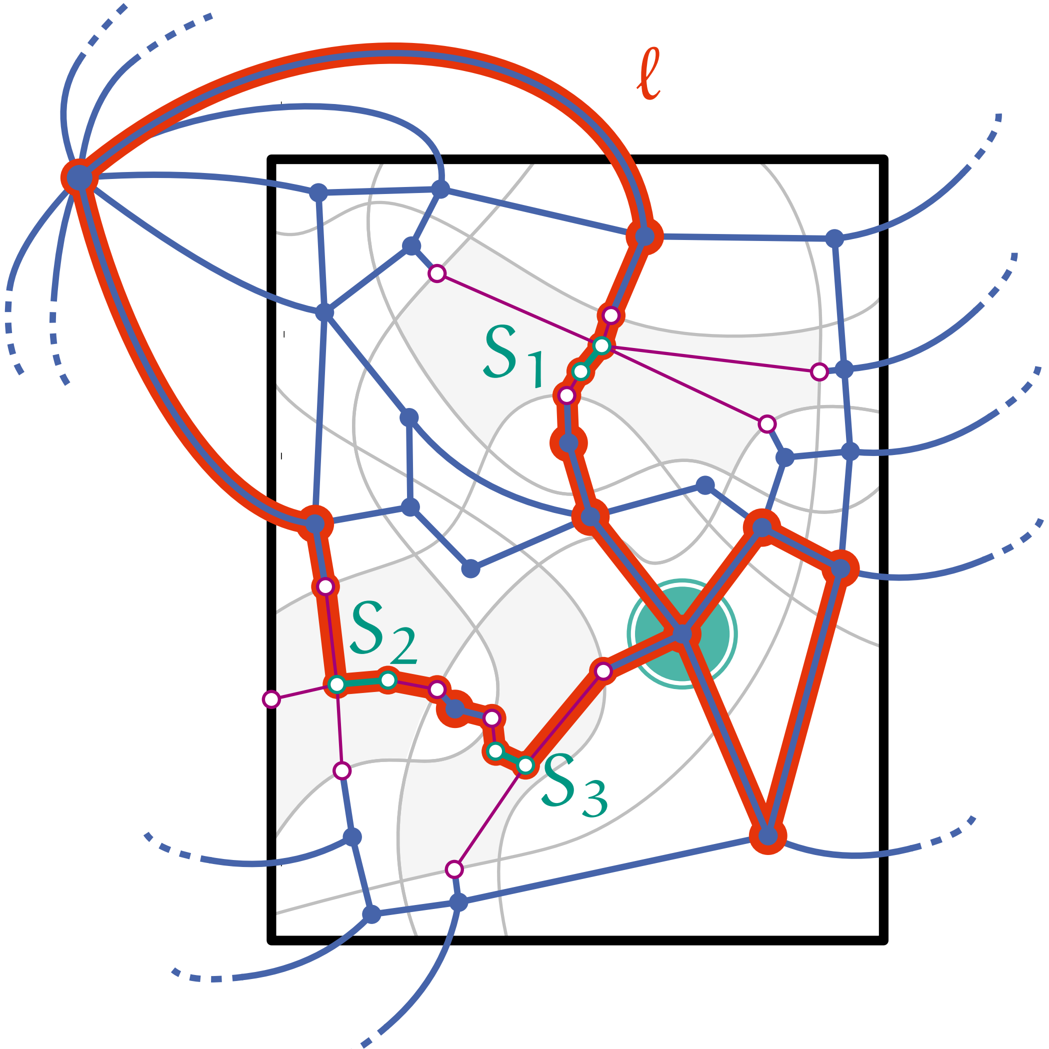
Curve arrangement $\mathcal{A}$

Additional curve c

Dual graph $\mathcal{A}^{\mathrm{d}}$

closed walk ℓ

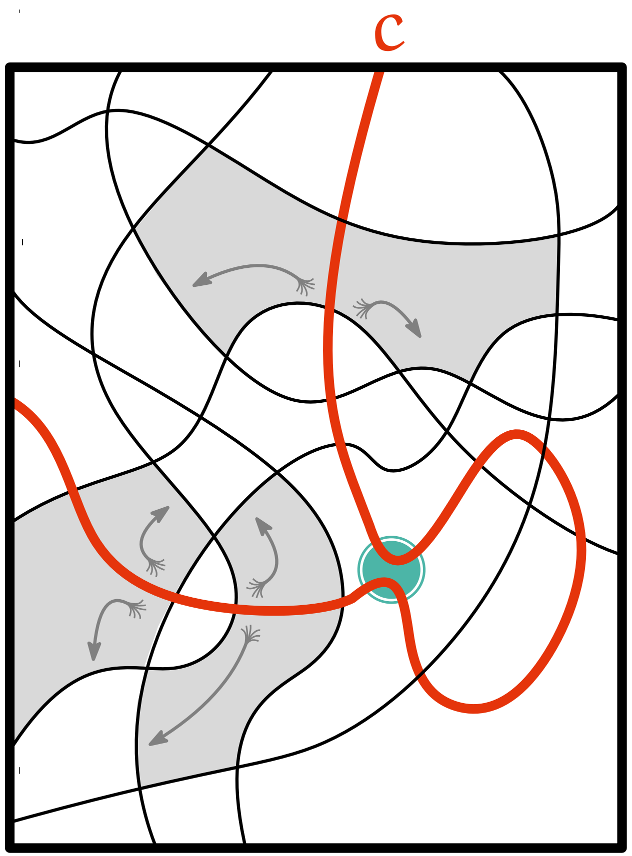




Representing a Resolution Curve

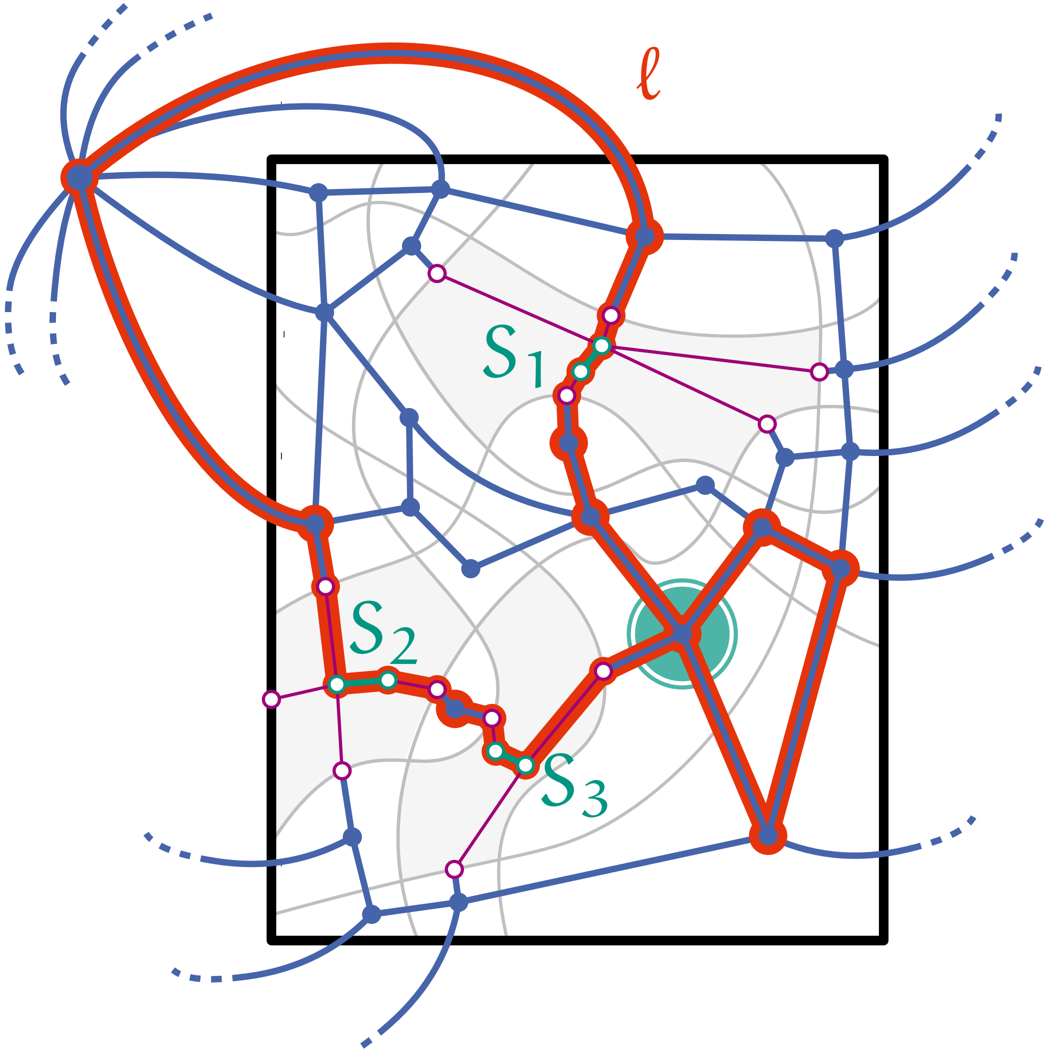
Curve arrangement $\mathcal{A}$

Additional curve c



Dual graph $\mathcal{A}^d$

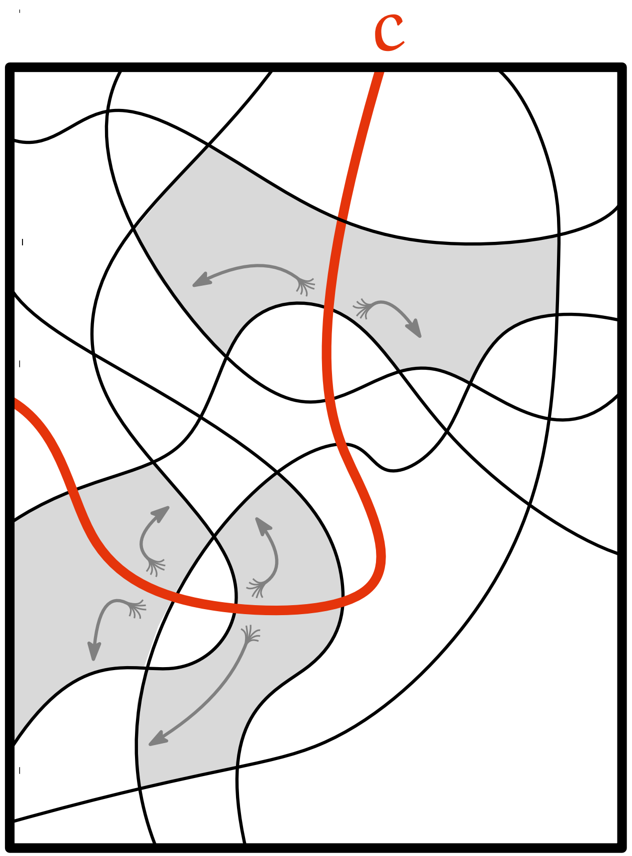
closed walk ℓ



Representing a Resolution Curve

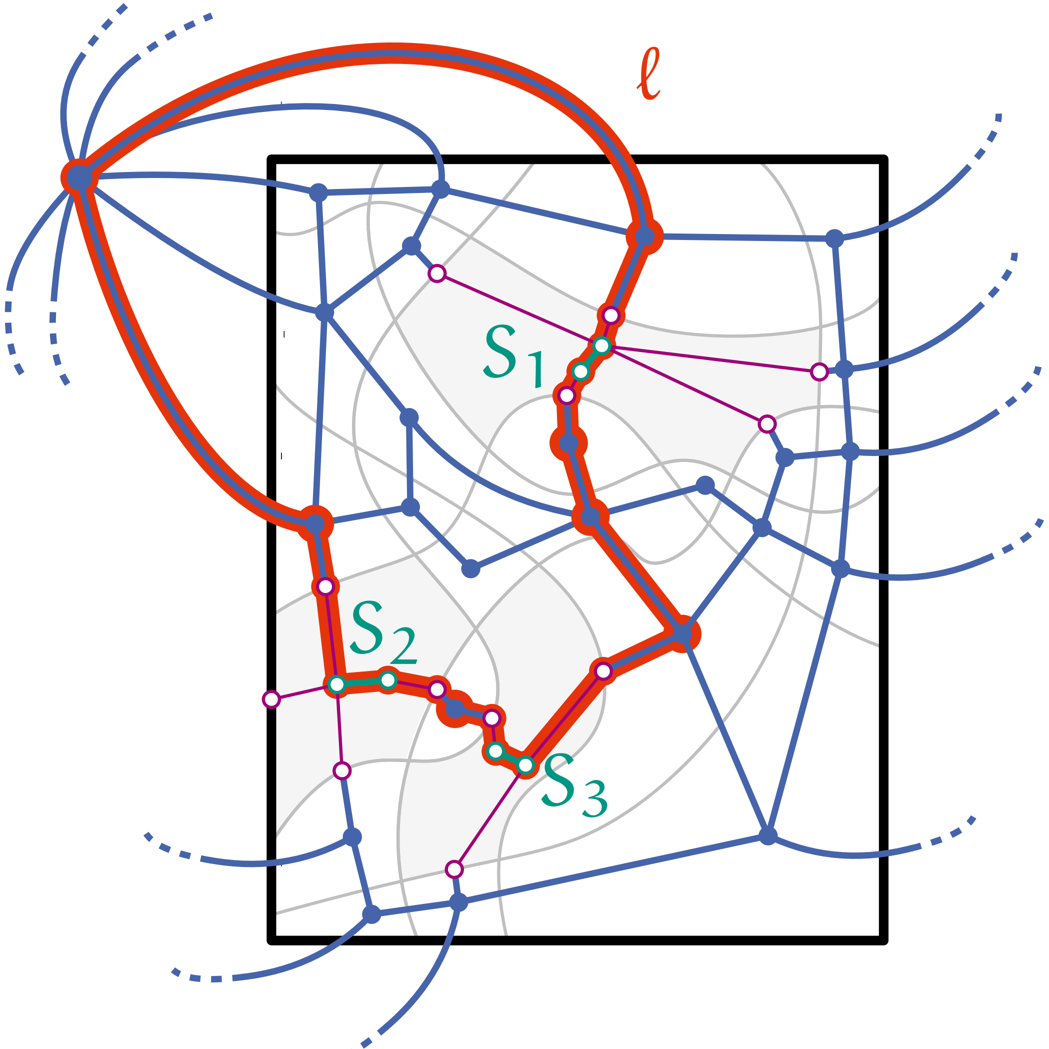
Curve arrangement $\mathcal{A}$

Additional curve c



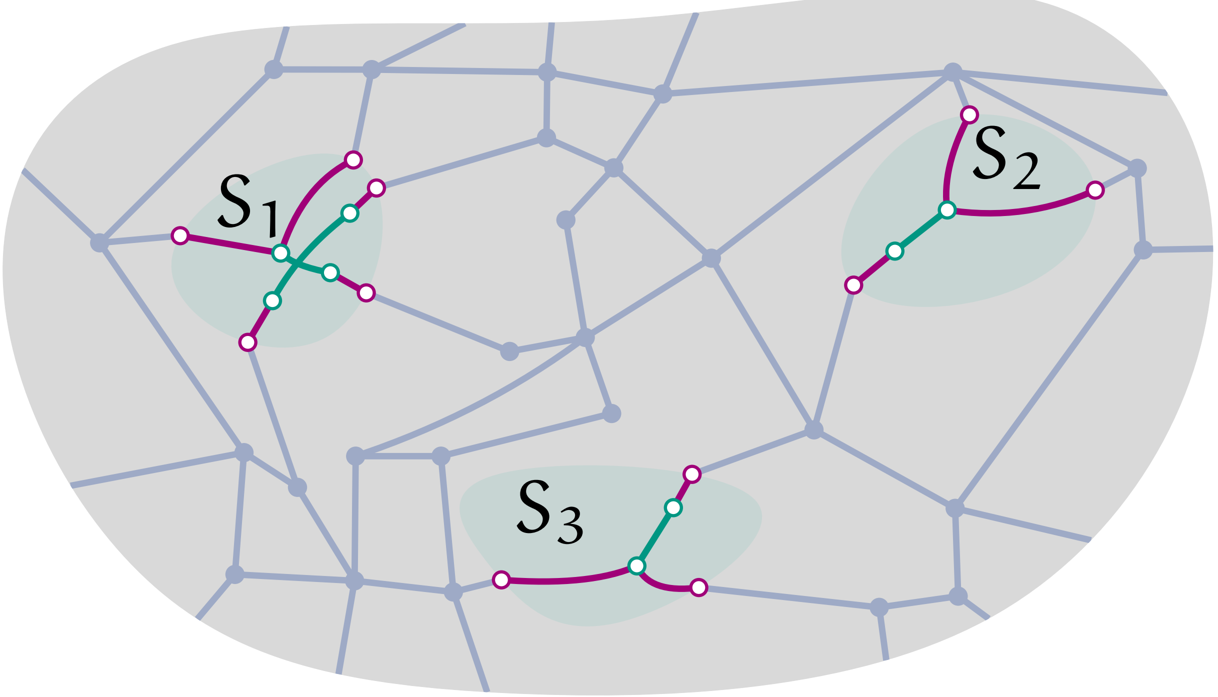
Dual graph $\mathcal{A}^{\mathrm{d}}$

Simple closed walk ℓ



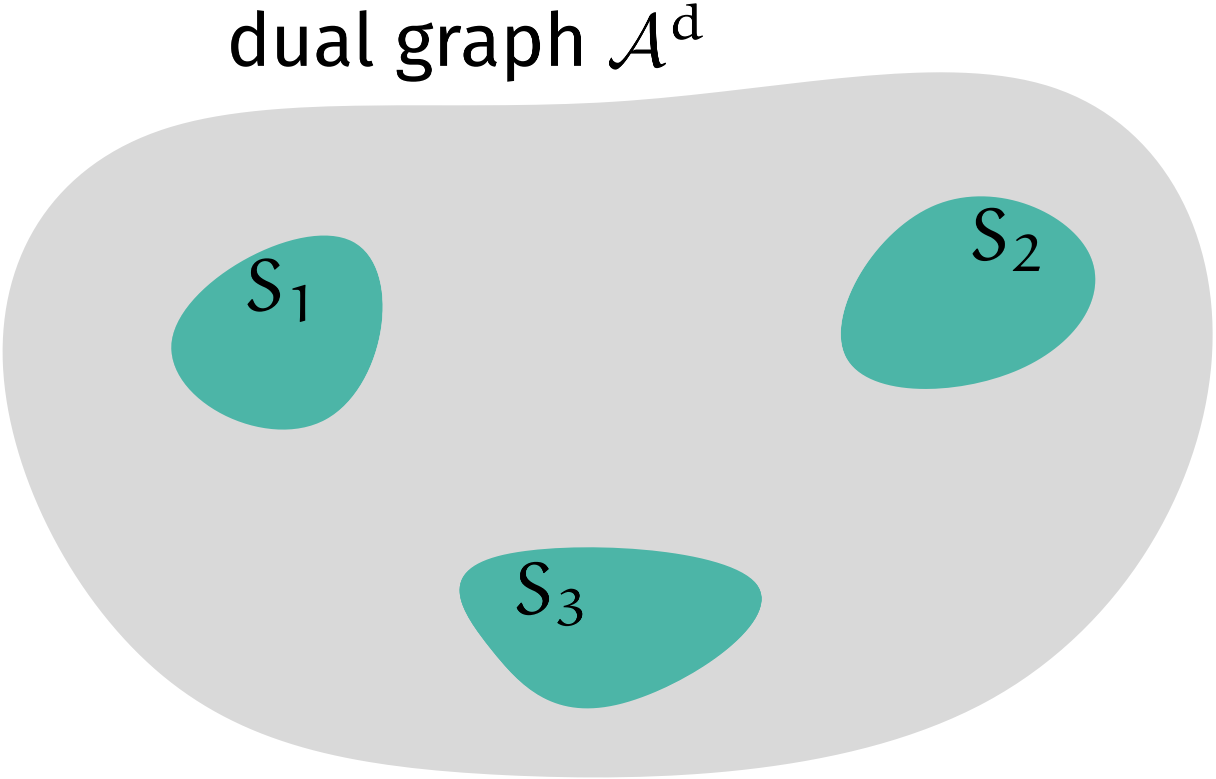
Computing Walks in Dual Graph with Dynamic Program

dual graph $\mathcal{A}^{\mathrm{d}}$



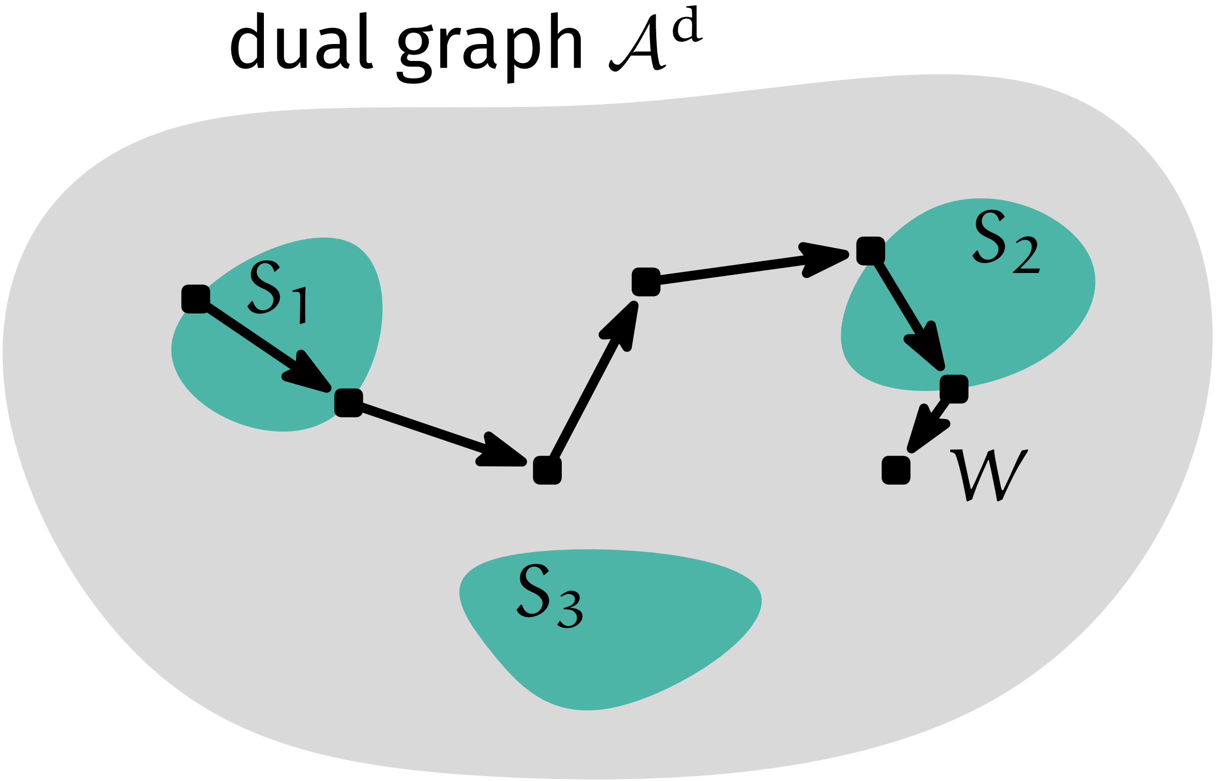
$\mathcal{S} = \{S_1, S_2, S_3\}$

Computing Walks in Dual Graph with Dynamic Program



$\mathcal{S} = \{S_1, S_2, S_3\}$

Computing Walks in Dual Graph with Dynamic Program

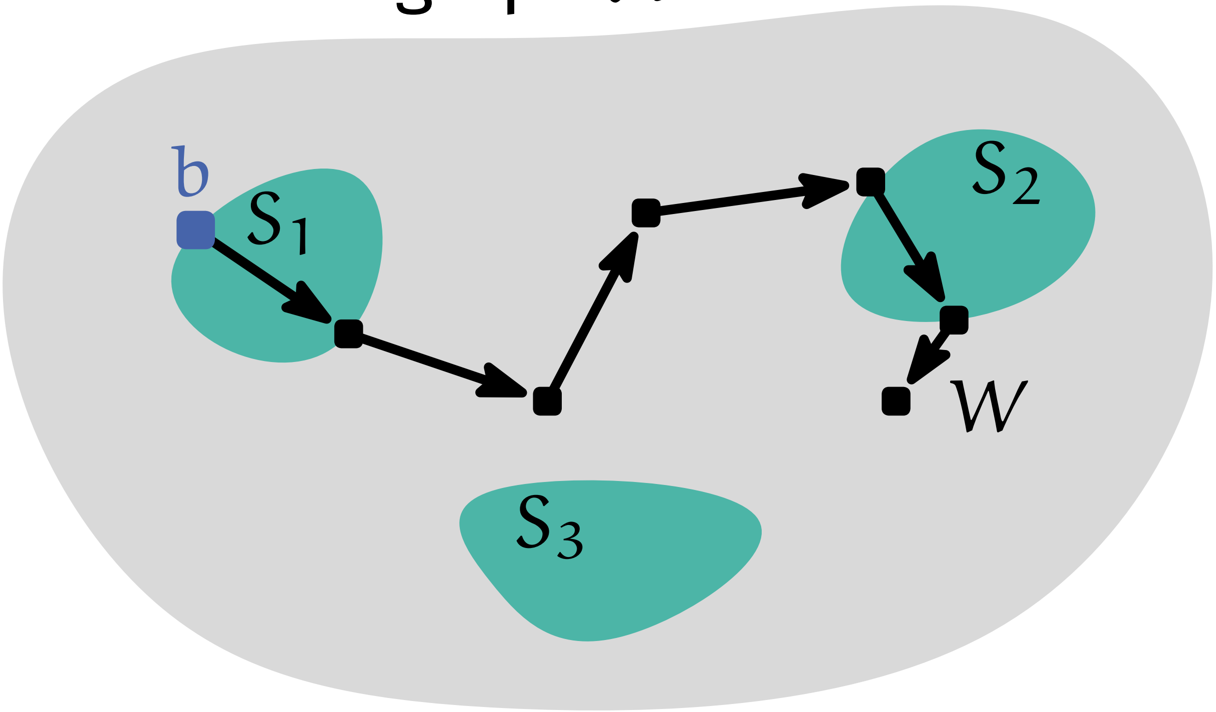


$\mathcal{S} = \{S_1, S_2, S_3\}$

Partial solution:
Walk W

Computing Walks in Dual Graph with Dynamic Program

dual graph $\mathcal{A}^{\mathrm{d}}$

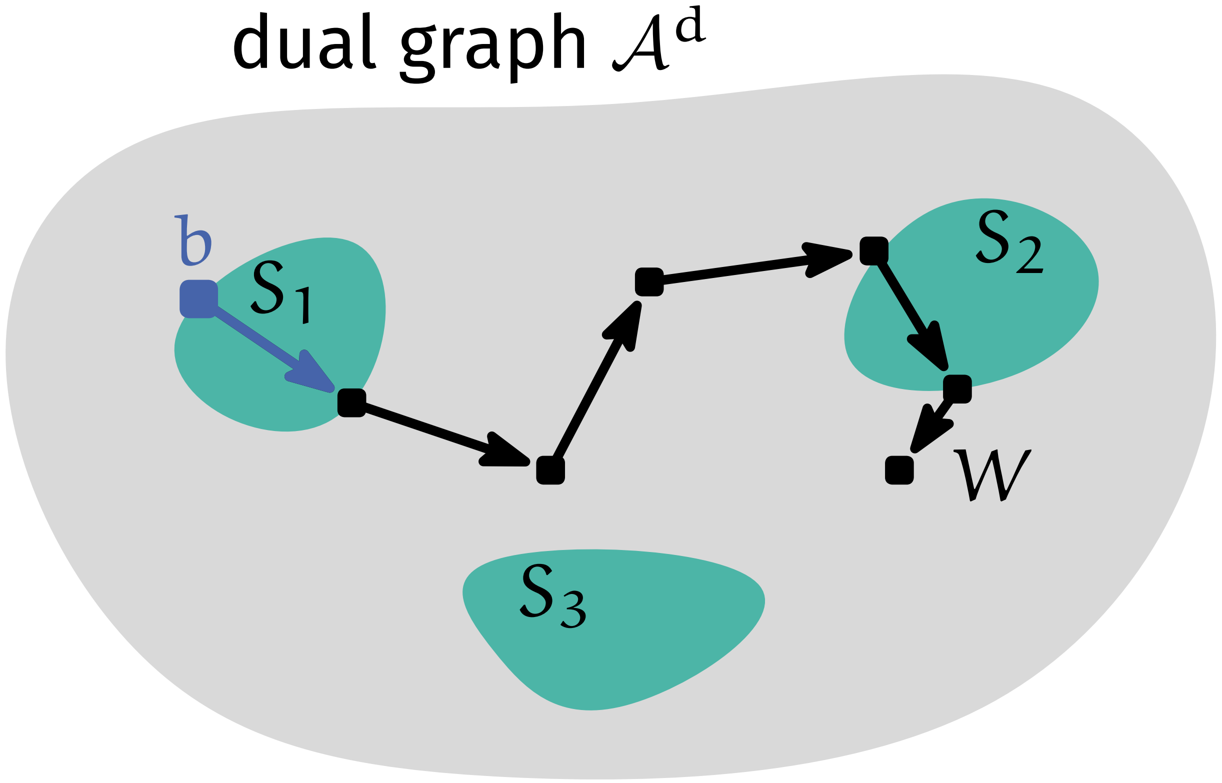


$\mathcal{S} = \{S_1, S_2, S_3\}$

Partial solution:
Walk W

- start at b

Computing Walks in Dual Graph with Dynamic Program

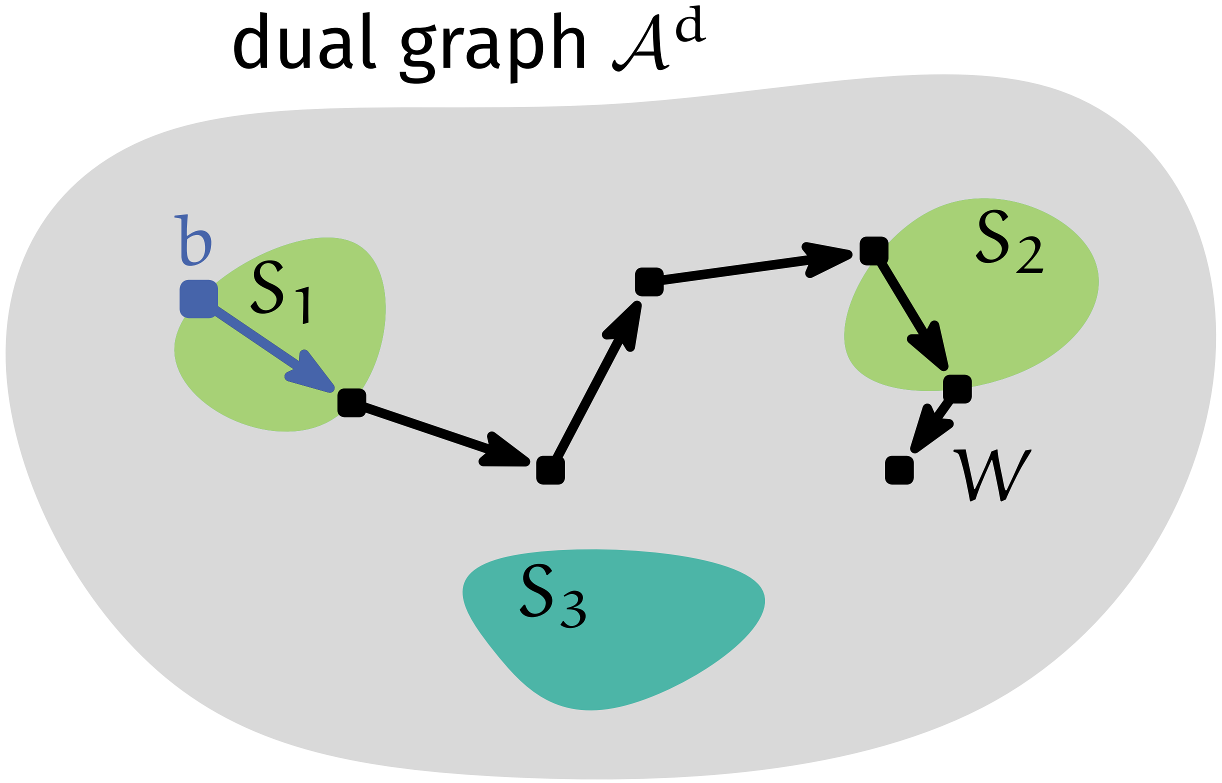


$\mathcal{S} = \{S_1, S_2, S_3\}$

Partial solution:
Walk W

- start at b
- first edge in S_1

Computing Walks in Dual Graph with Dynamic Program



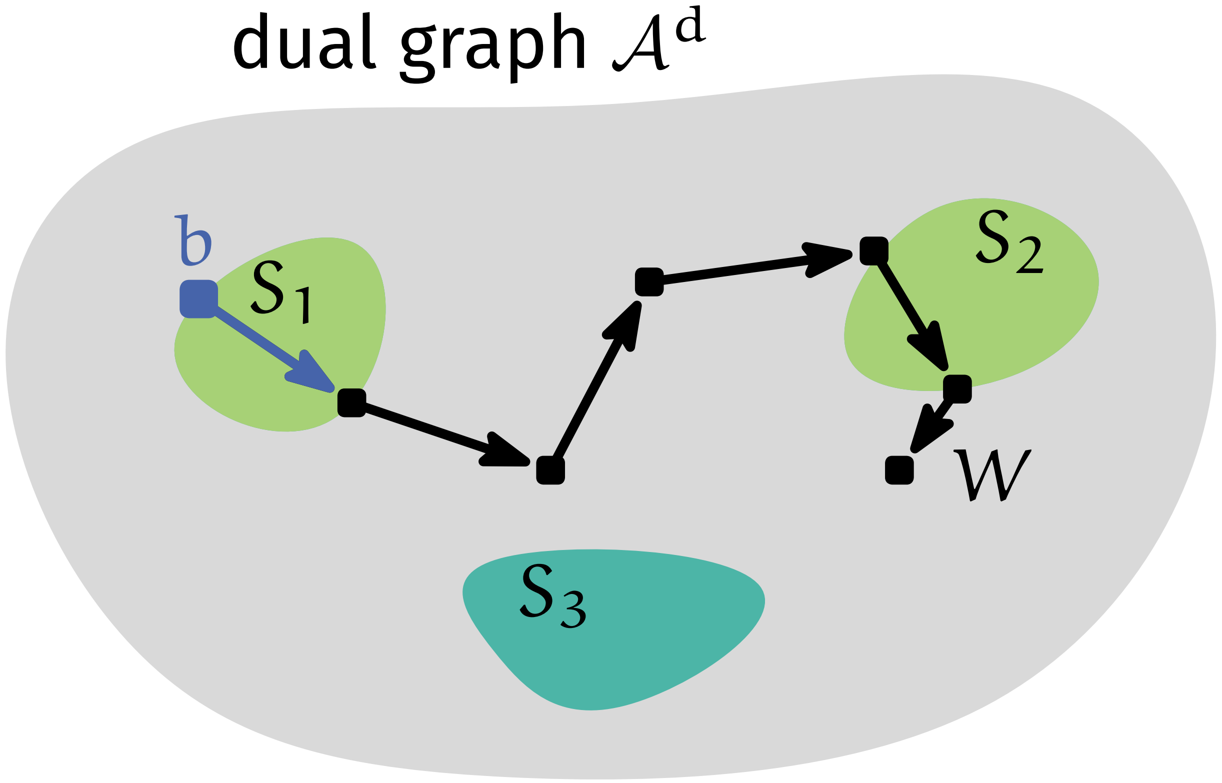
R

$\mathcal{S} = \{S_1, S_2, S_3\}$

Partial solution:
Walk W

- start at b
- first edge in S_1
- cross exactly R $\subseteq \mathcal{S}$

Computing Walks in Dual Graph with Dynamic Program

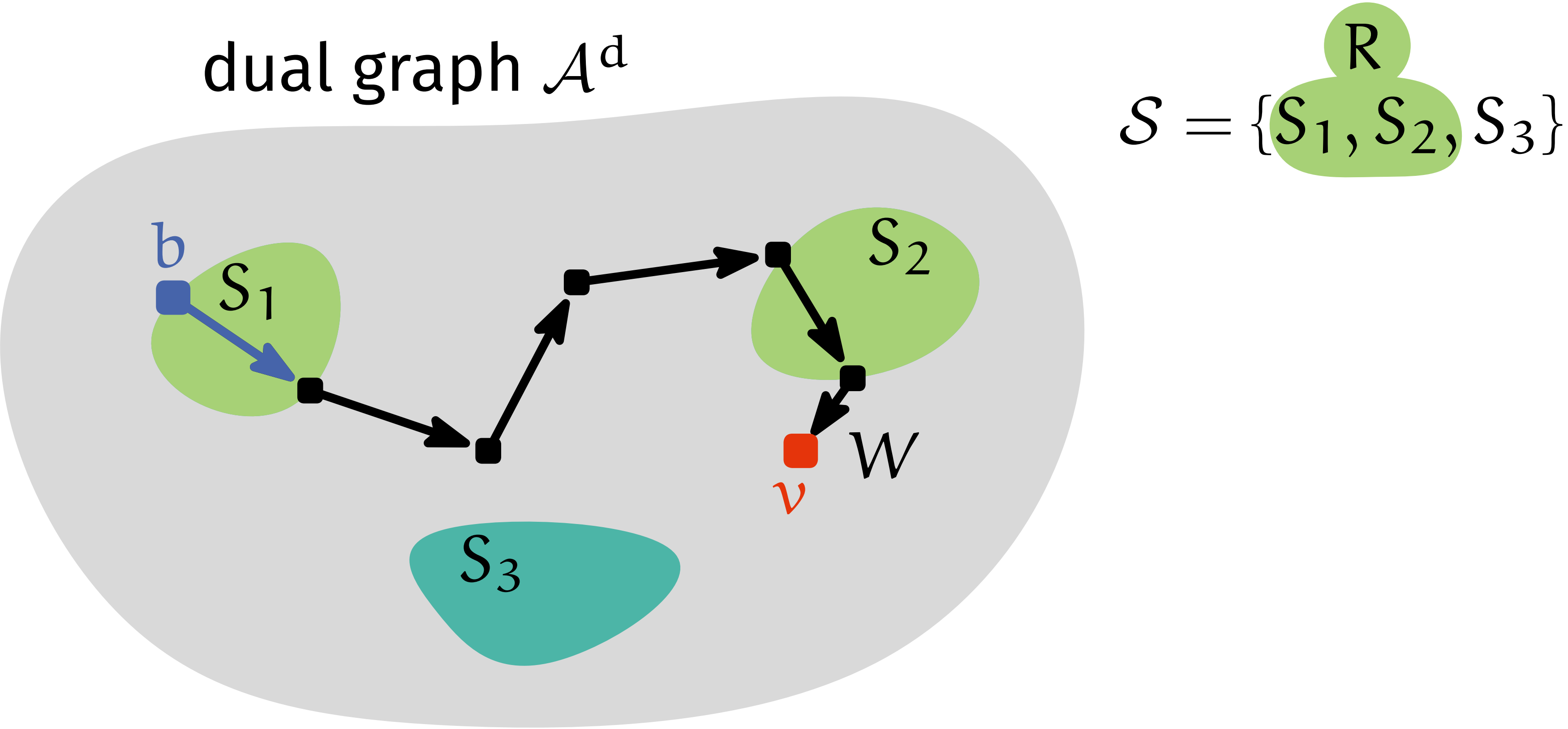


$$\mathcal{S} = \{\overset{R}{S_1, S_2}, S_3\}$$

Partial solution:
Walk W

- start at b
- first edge in S_1
- cross exactly $R \subseteq \mathcal{S}$
- m edges long

Computing Walks in Dual Graph with Dynamic Program

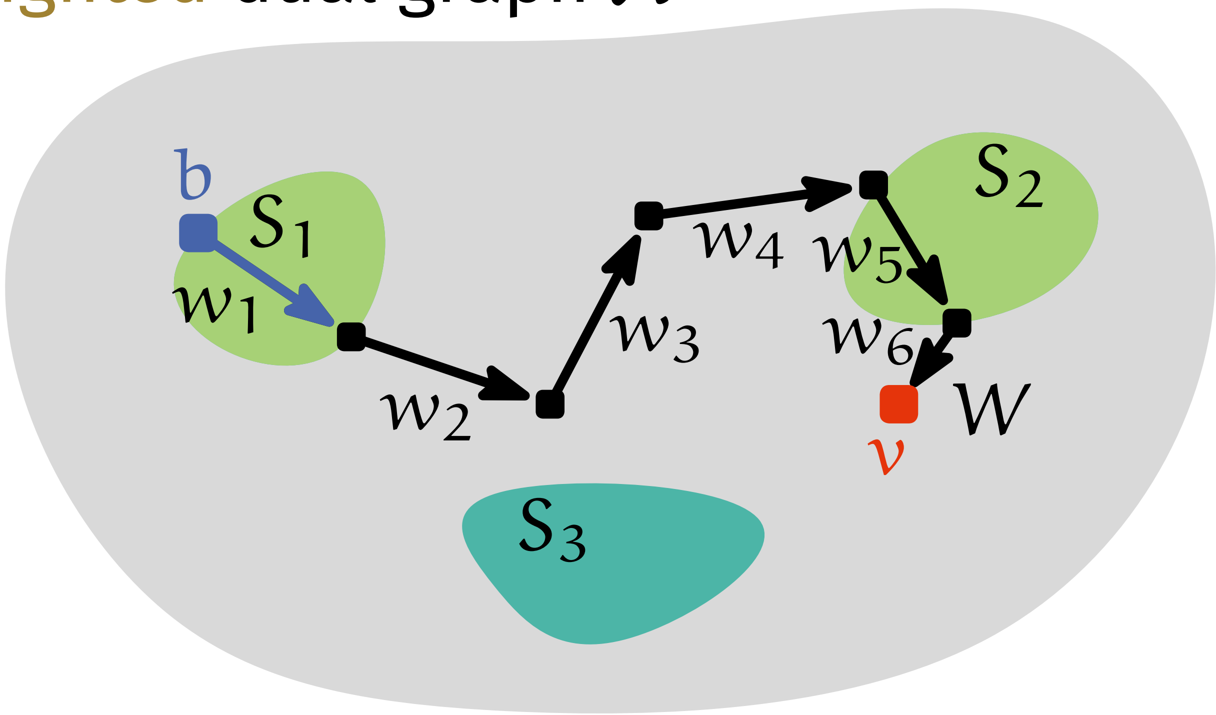


Partial solution:
Walk W

- start at b
- first edge in S_1
- cross exactly $R \subseteq \mathcal{S}$
- m edges long
- end at v

Computing Walks in Dual Graph with Dynamic Program

Weighted dual graph $\mathcal{A}^d$



$\mathcal{S} = \{S_1, S_2, S_3\}$, with R marking S_1, S_2

Partial solution:
Walk W

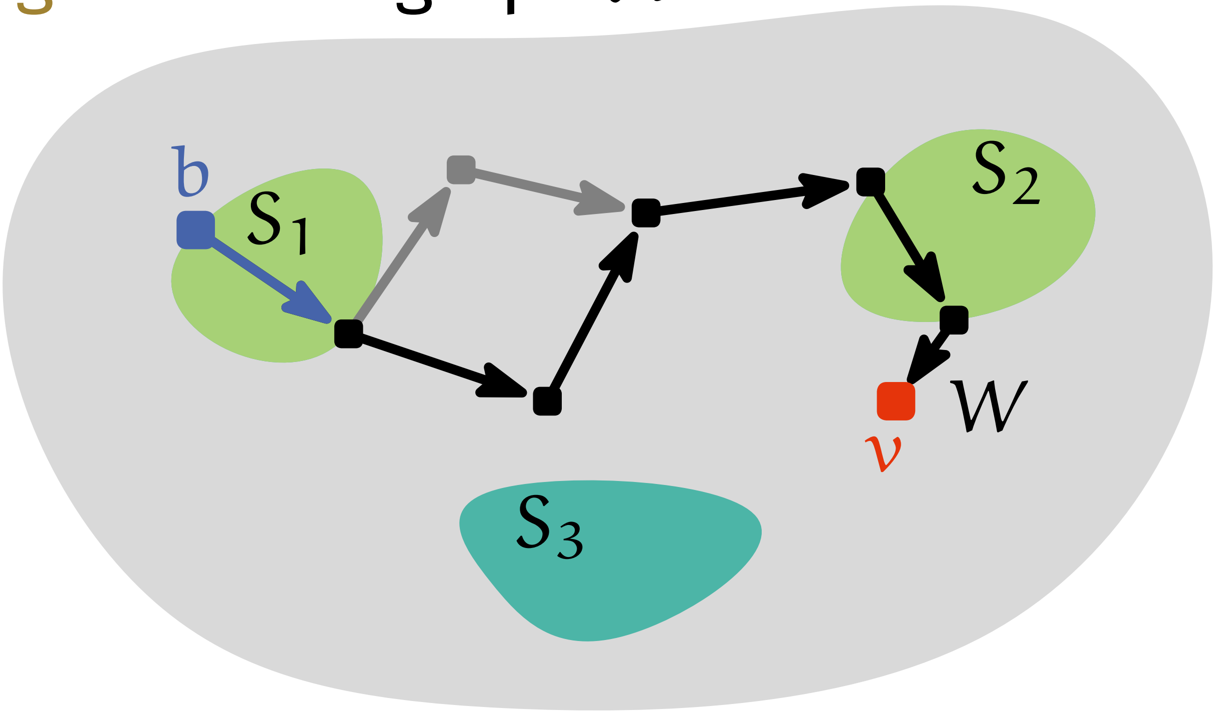
Representing W as single value:

$$f(W) = \prod_{i=1}^{m} w_i$$

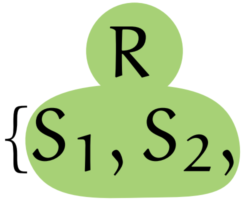
- start at b
- first edge in S_1
- cross exactly $R \subseteq \mathcal{S}$
- m edges long
- end at v

Computing Walks in Dual Graph with Dynamic Program

Weighted dual graph $\mathcal{A}^d$



$\mathcal{S} = \{S_1, S_2, S_3\}$



Partial solution:
Walk W

Representing W as single value:

$$f(W) = \prod_{i=1}^{m} w_i$$

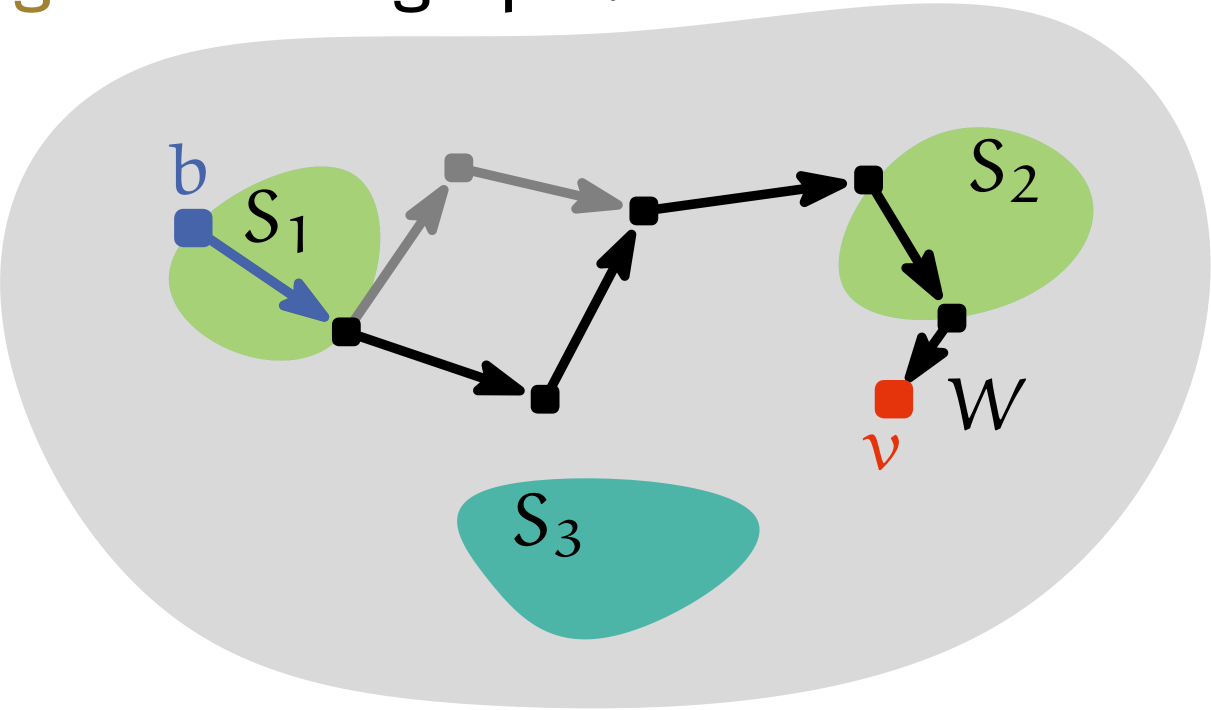
Set Ω of walks in $\mathcal{A}^d$

- start at b
- first edge in S_1
- cross exactly $R \subseteq \mathcal{S}$
- m edges long
- end at v

There are multiple such walks!

Computing Walks in Dual Graph with Dynamic Program

Weighted dual graph $\mathcal{A}^d$



$\mathcal{S} = \{S_1, S_2, S_3\}$ with R marking $\{S_1, S_2\}$

Partial solution:
Walk W

Representing W as single value:

$$f(W) = \prod_{i=1}^{m} w_i$$

Representing Ω as a single value:

$$T_b(R, m, v) = \sum_{W \in \Omega} f(W)$$

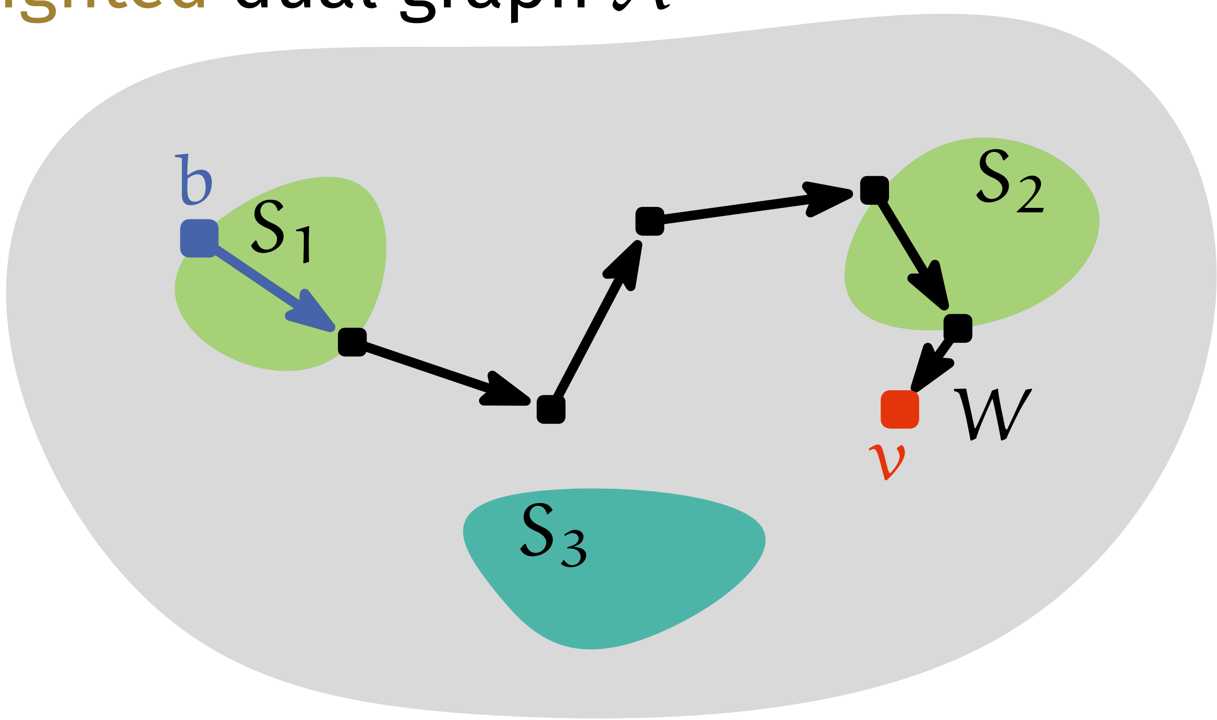
Set Ω of walks in $\mathcal{A}^d$

- start at b
- first edge in S_1
- cross exactly $R \subseteq \mathcal{S}$
- m edges long
- end at v

There are multiple such walks!

Computing Walks in Dual Graph with Dynamic Program

Weighted dual graph $\mathcal{A}^d$



$\mathcal{S} = \{\overset{R}{S_1, S_2}, S_3\}$

Partial solution:
Walk W

Representing W as single value:

$$f(W) = \prod_{i=1}^{m} w_i$$

Representing Ω as a single value:

$$T_b(R, m, v) = \sum_{W \in \Omega} f(W)$$

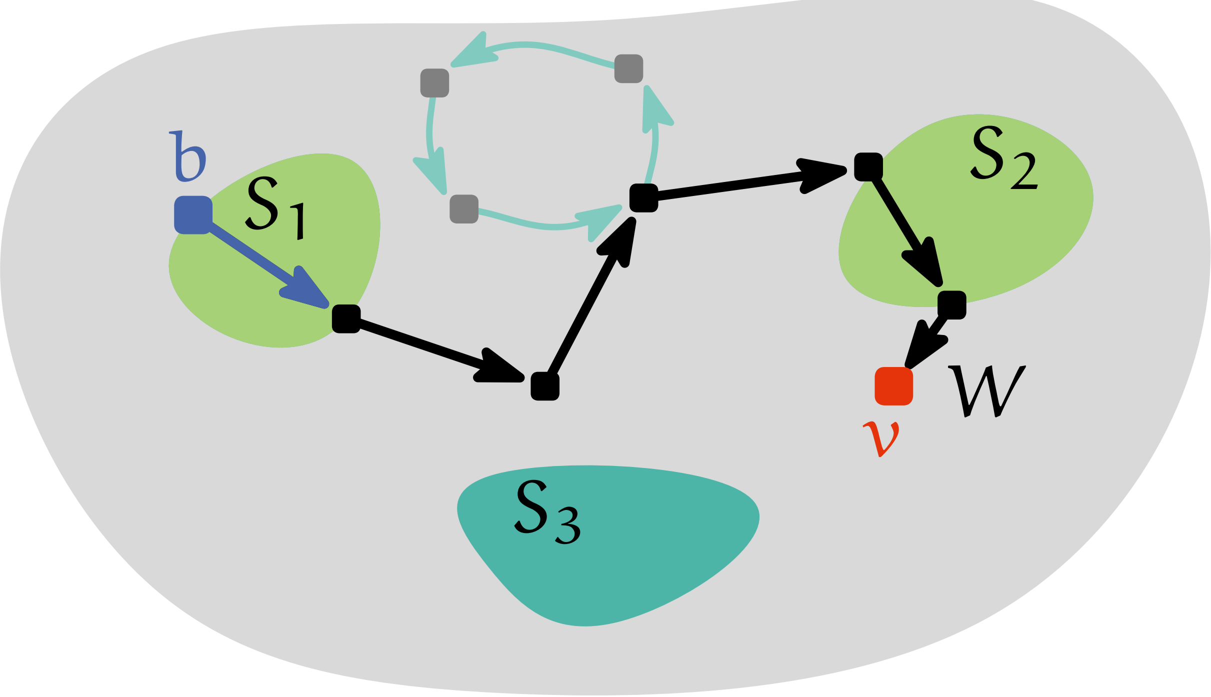
Set Ω of walks in $\mathcal{A}^d$

- start at b
- first edge in S_1
- cross exactly $R \subseteq \mathcal{S}$
- m edges long
- end at v

We can tell if there is a simple walk in Ω
(with high certainty)

Computing Walks in Dual Graph with Dynamic Program

Weighted dual graph $\mathcal{A}^d$



$\mathcal{S} = \{\underbrace{S_1, S_2,}_{} S_3\}$ R

Partial solution:
Walk W

Representing W as single value:

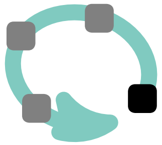
$$f(W) = \prod_{i=1}^{m} w_i$$

Representing Ω as a single value:

$$T_b(R, m, v) = \sum_{W \in \Omega} f(W)$$

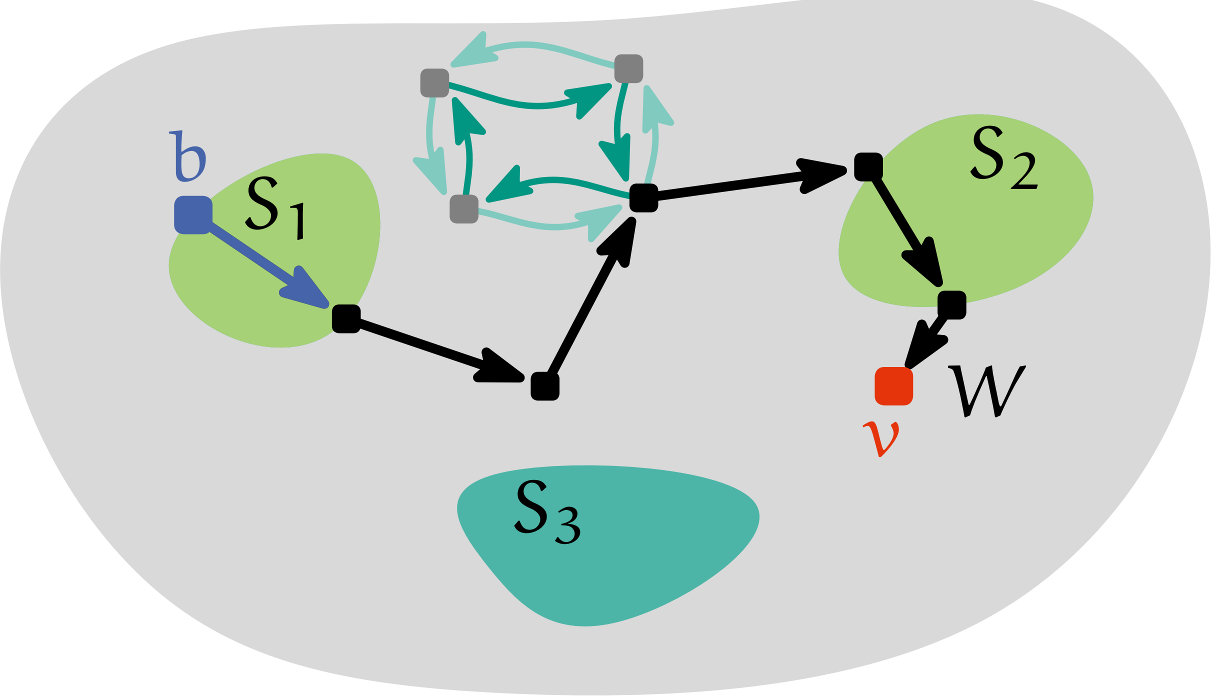
Set Ω of walks in $\mathcal{A}^d$

- start at b
- first edge in S_1
- cross exactly $R \subseteq \mathcal{S}$
- m edges long
- end at v

f()

Computing Walks in Dual Graph with Dynamic Program

Weighted dual graph $\mathcal{A}^d$



$\mathcal{S} = \{\overbrace{S_1, S_2}^{R}, S_3\}$

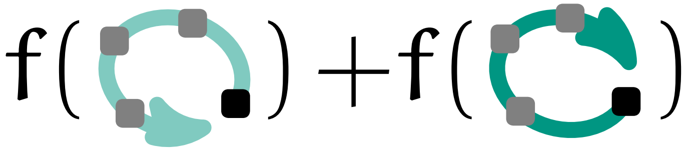
Partial solution:
Walk W

Representing W as single value:

$$f(W) = \prod_{i=1}^{m} w_i$$

Representing Ω as a single value:

$$T_b(R, m, v) = \sum_{W \in \Omega} f(W)$$

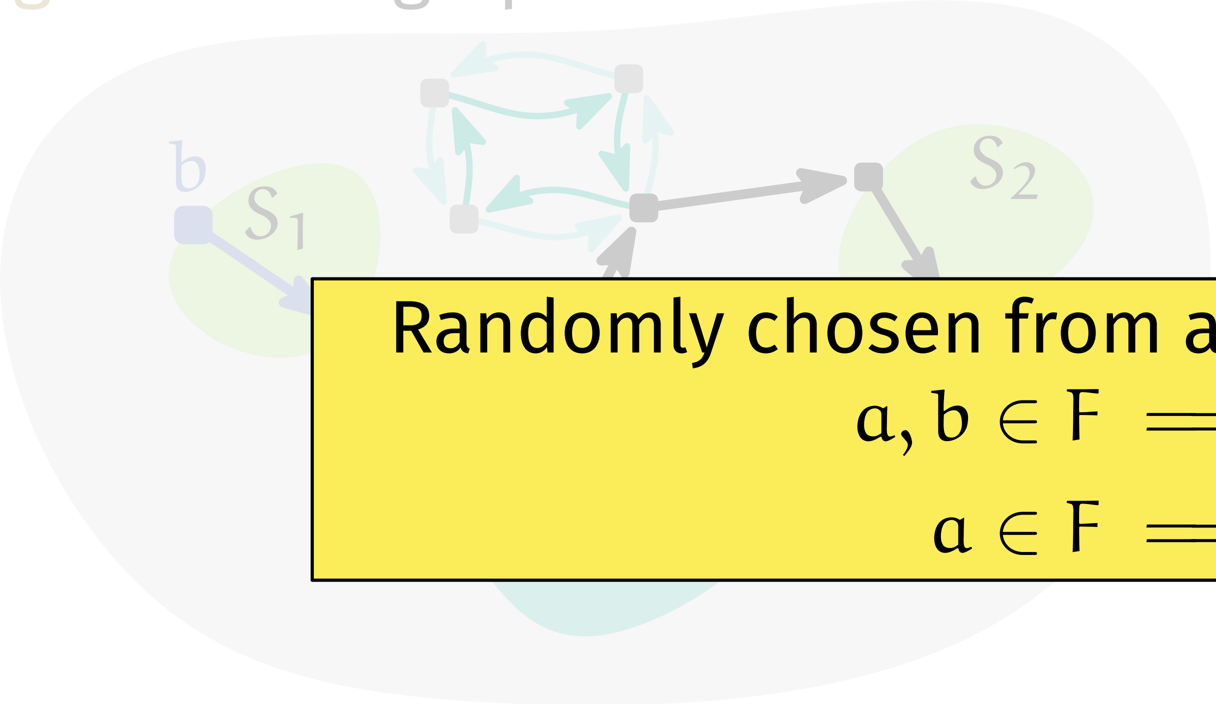


Set Ω of walks in $\mathcal{A}^d$

- start at b
- first edge in S_1
- cross exactly $R \subseteq \mathcal{S}$
- m edges long
- end at v

Computing Walks in Dual Graph with Dynamic Program

Weighted dual graph $\mathcal{A}^d$



$\mathcal{S} = \{S_1, S_2, S_3\}$ with R marked

Partial solution:
Walk W

Representing W as single value:

Randomly chosen from a Field F of characteristic 2

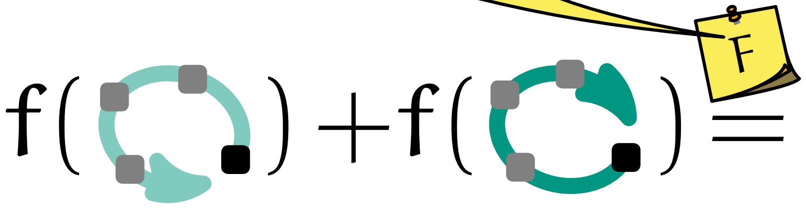
$$a, b \in F \implies a \cdot b \in F$$

$$a \in F \implies a + a = 0$$

value:

Set Ω of walks in $\mathcal{A}^d$

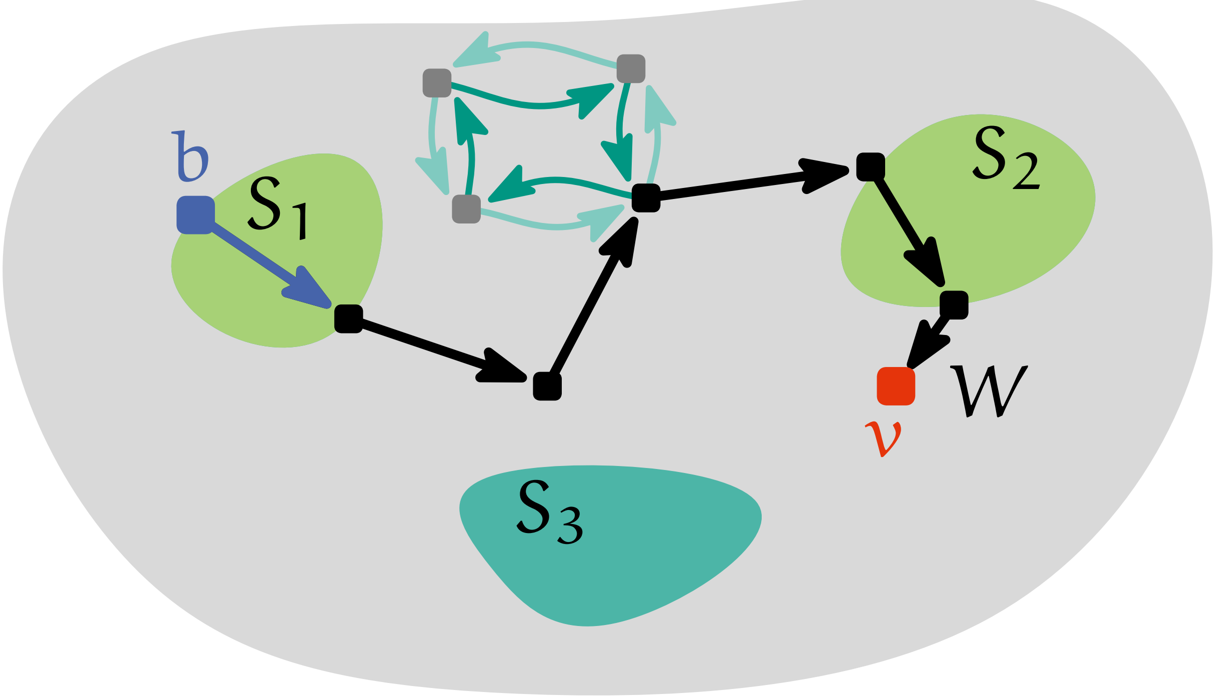
- start at b
- first edge in S_1
- cross exactly $R \subseteq \mathcal{S}$
- m edges long
- end at v



[Björklund et al., ’12]

Computing Walks in Dual Graph with Dynamic Program

Weighted dual graph $\mathcal{A}^d$



Set Ω of walks in $\mathcal{A}^d$

- start at b
- first edge in S_1
- cross exactly $R \subseteq \mathcal{S}$
- m edges long
- end at v

$$\mathcal{S} = \{\overset{R}{S_1, S_2}, S_3\}$$

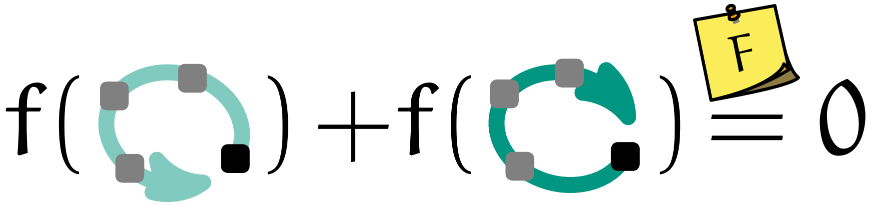
Partial solution:
Walk W

Representing W as single value:

$$f(W) = \prod_{i=1}^{m} w_i$$

Representing Ω as a single value:

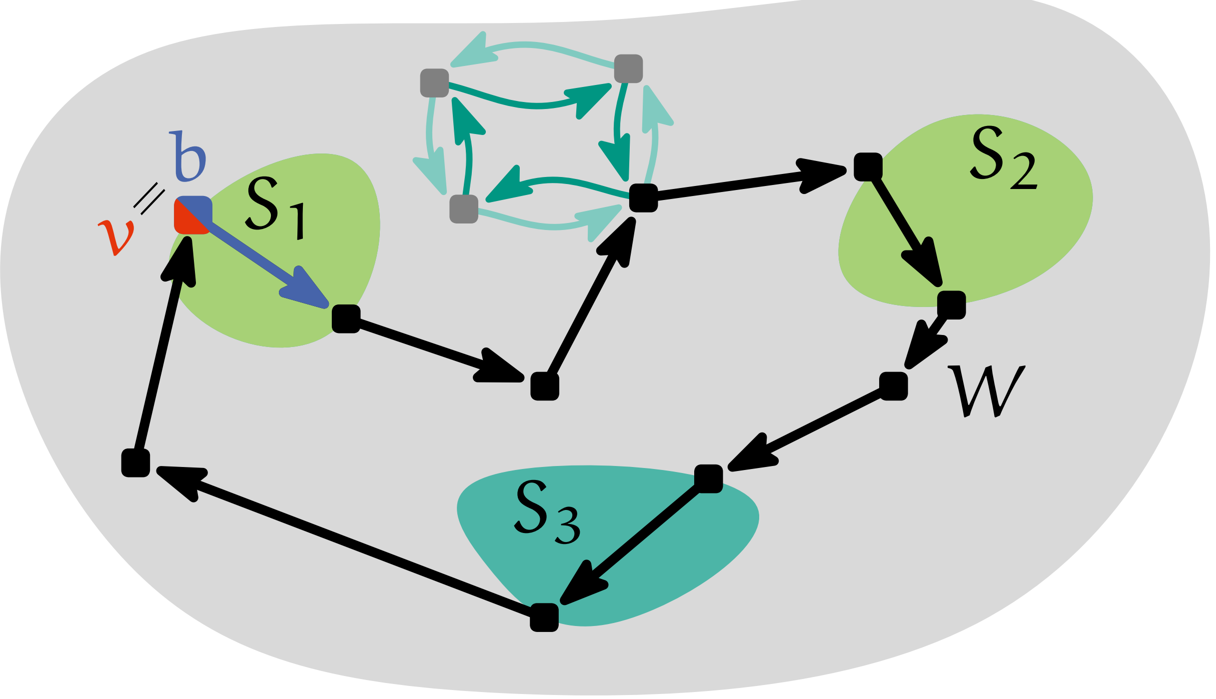
$$T_b(R, m, v) = \sum_{W \in \Omega} f(W)$$



[Björklund et al., '12]

Computing Walks in Dual Graph with Dynamic Program

Weighted dual graph $\mathcal{A}^d$



Set Ω of walks in $\mathcal{A}^d$

- start at b
- first edge in S_1
- cross exactly $R \subseteq \mathcal{S}$
- m edges long
- end at v

$$\mathcal{S} = \{\overbrace{S_1, S_2,}^{R} S_3\}$$

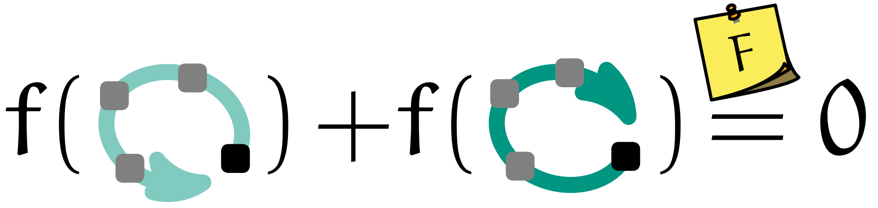
Partial solution:
Walk W

Representing W as single value:

$$f(W) = \prod_{i=1}^{m} w_i$$

Representing Ω as a single value:

$$T_b(R, m, v) = \sum_{W \in \Omega} f(W)$$

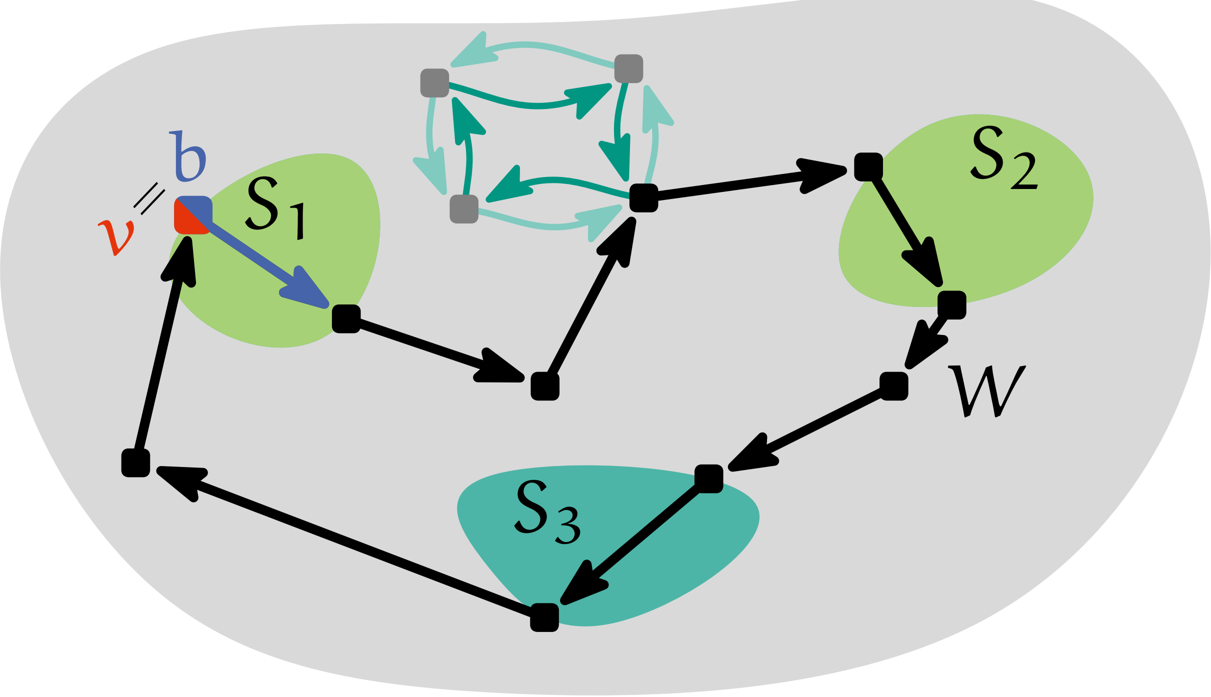


[Björklund et al., ’12]

$$T_b(\mathcal{S}, 1, b) = 0$$

Computing Walks in Dual Graph with Dynamic Program

Weighted dual graph $\mathcal{A}^d$



Set Ω of walks in $\mathcal{A}^d$

- start at b
- first edge in S_1
- cross exactly $R \subseteq \mathcal{S}$
- m edges long
- end at v

$$\mathcal{S} = \{\overset{R}{S_1, S_2}, S_3\}$$

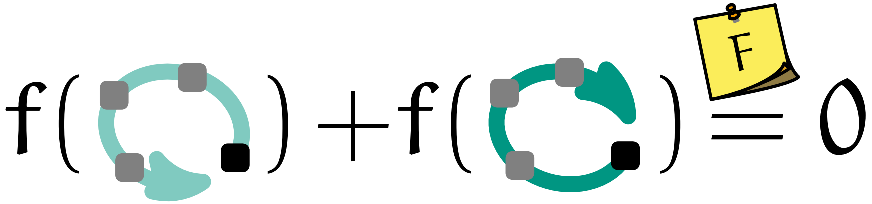
Partial solution:
Walk W

Representing W as single value:

$$f(W) = \prod_{i=1}^{m} w_i$$

Representing Ω as a single value:

$$T_b(R, m, v) = \sum_{W \in \Omega} f(W)$$

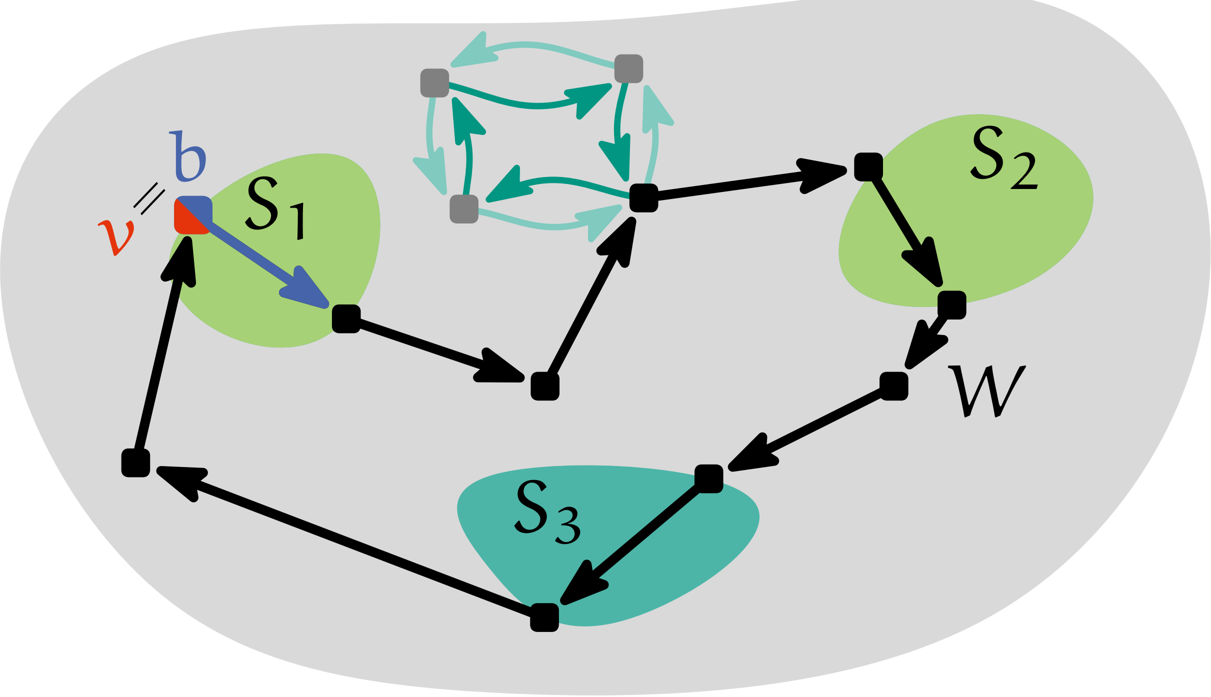


[Björklund et al., ’12]

$$T_b(\mathcal{S}, 2, b) = 0$$

Computing Walks in Dual Graph with Dynamic Program

Weighted dual graph $\mathcal{A}^d$



$$\mathcal{S}=\{\overbrace{S_1,S_2,}^{R}S_3\}$$

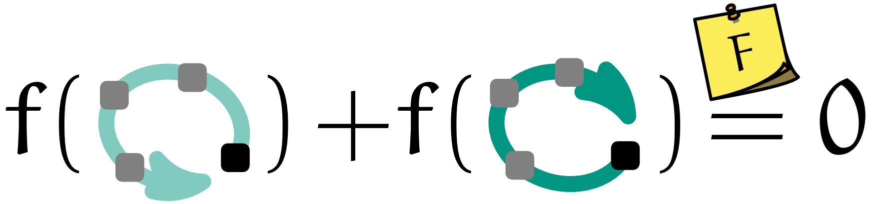
Partial solution:
Walk W

Representing W as single value:

$$f(W)=\prod_{i=1}^{m} w_i$$

Representing Ω as a single value:

$$T_b(R,m,v)=\sum_{W\in\Omega} f(W)$$



[Björklund et al., ’12]

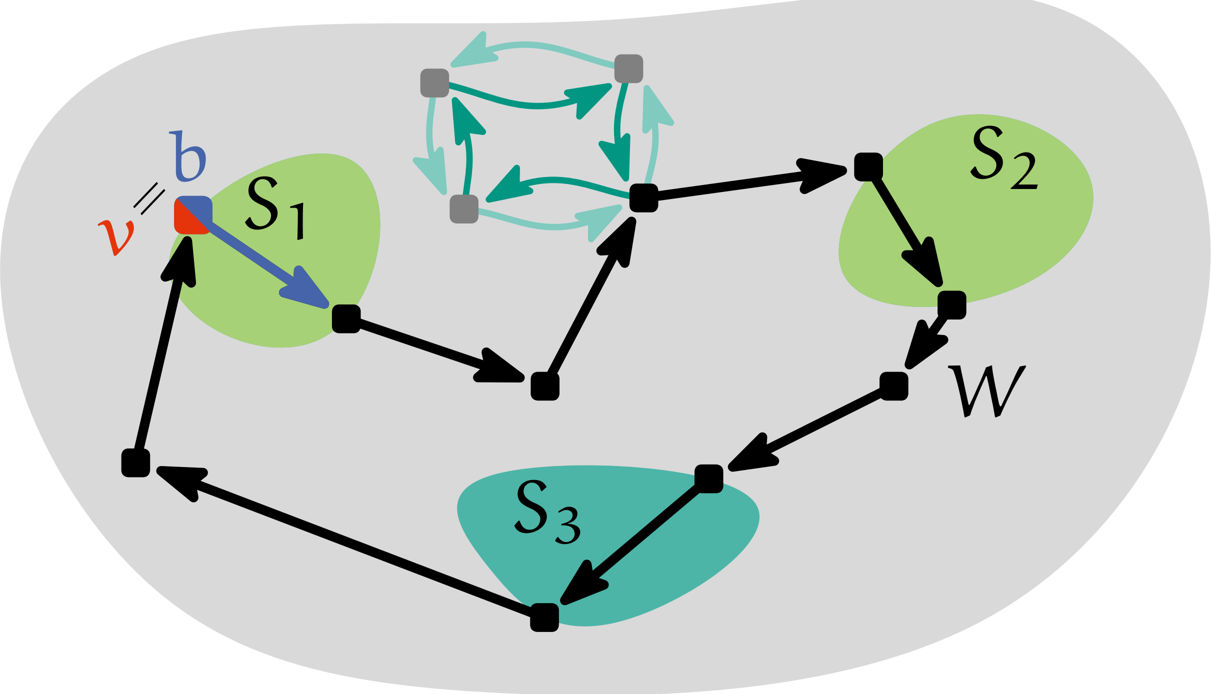
$$T_b(\mathcal{S},3,b)=0$$

Set Ω of walks in $\mathcal{A}^d$

- start at b
- first edge in S_1
- cross exactly $R \subseteq \mathcal{S}$
- m edges long
- end at v

Computing Walks in Dual Graph with Dynamic Program

Weighted dual graph $\mathcal{A}^d$



$$\mathcal{S} = \{\overset{R}{S_1, S_2}, S_3\}$$

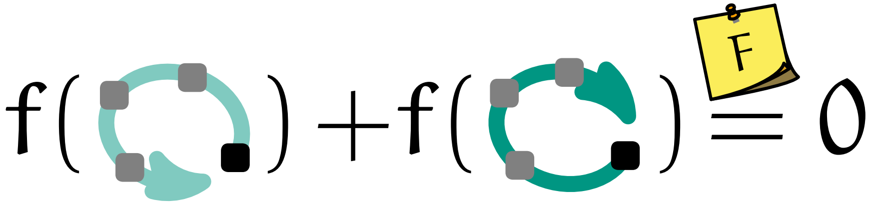
Partial solution:
Walk W

Representing W as single value:

$$f(W) = \prod_{i=1}^{m} w_i$$

Representing Ω as a single value:

$$T_b(R, m, v) = \sum_{W \in \Omega} f(W)$$



[Björklund et al., ’12]

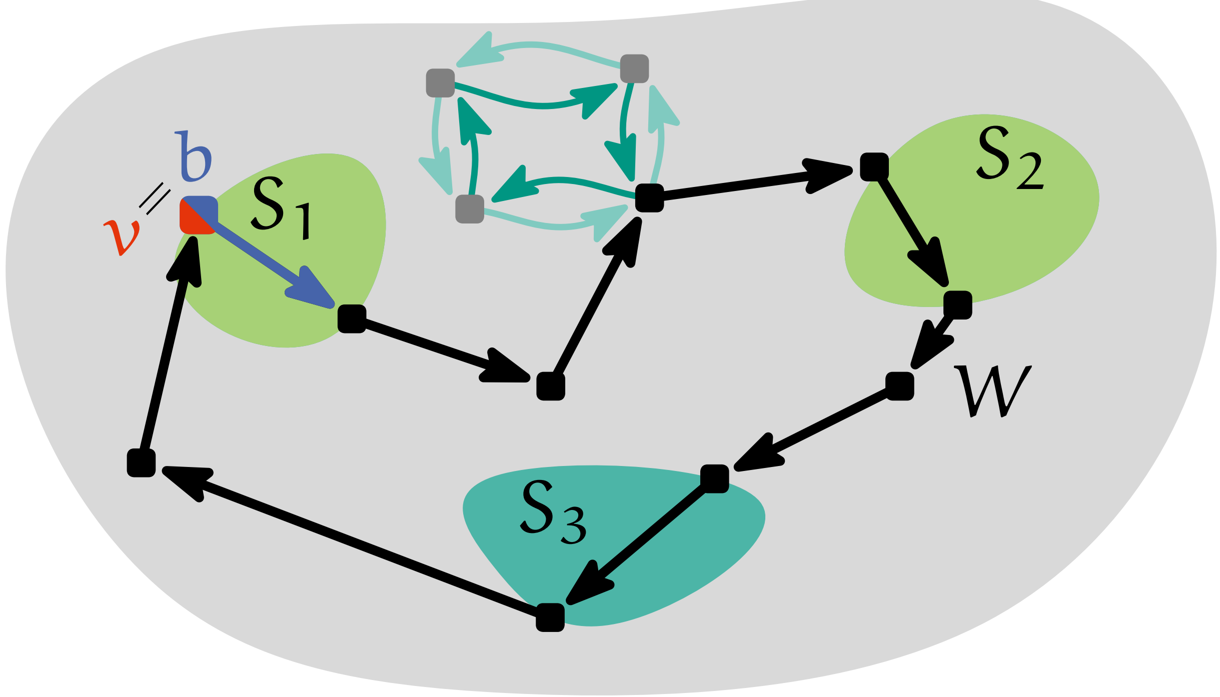
Set Ω of walks in $\mathcal{A}^d$

- start at b
- first edge in S_1
- cross exactly $R \subseteq \mathcal{S}$
- m edges long
- end at v

∃ simple walk of length m' $\Leftarrow$ $T_b(\mathcal{S}, m', b) \neq 0$ for some m'

Computing Walks in Dual Graph with Dynamic Program

Weighted dual graph $\mathcal{A}^d$



$\mathcal{S} = \{S_1, S_2, S_3\}$ R

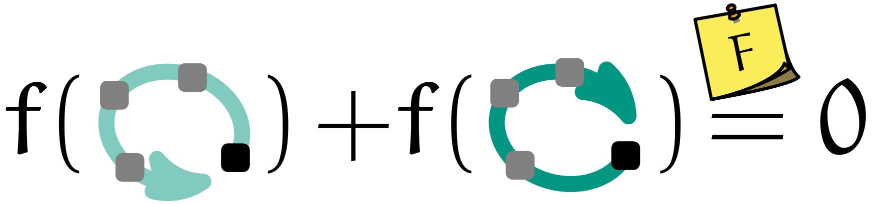
Partial solution:
Walk W

Representing W as single value:

$$f(W) = \prod_{i=1}^{m} w_i$$

Representing Ω as a single value:

$$T_b(R, m, v) = \sum_{W \in \Omega} f(W)$$



[Björklund et al., ’12]

Set Ω of walks in $\mathcal{A}^d$

- start at b
- first edge in S_1
- cross exactly $R \subseteq \mathcal{S}$
- m edges long
- end at v

$\exists$ simple walk of length m' $\Longleftarrow$ $\cancel{\Longrightarrow}$ $T_b(\mathcal{S}, m', b) \neq 0$ for some m'

One-sided Error

Randomized FPT Runtime

Computing $T_b(\mathcal{S}, m', b)$ in a graph $G = (V, E)$ via dynamic program

can be done in $\mathcal{O}\left(2^k \cdot k \cdot W \cdot |V(S_1)| \cdot |E||V|^2 \cdot \log\left(\frac{2|E|}{|V|}\right)\right)$

adapted from [Björklund et al., ’12]

Randomized FPT Runtime

Computing $T_b(\mathcal{S}, m', b)$ in a graph $G = (V, E)$ via dynamic program

can be done in $\mathcal{O}\left(2^k \cdot k \cdot W \cdot |V(S_1)| \cdot |E||V|^2 \cdot \log\left(\frac{2|E|}{|V|}\right)\right)$

adapted from [Björklund et al., ’12]

k := number of sets $S_1, \ldots, S_k$

Randomized FPT Runtime

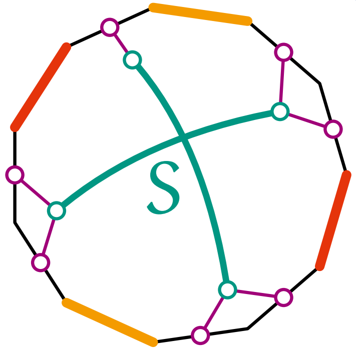
Computing $T_b(\mathcal{S}, m', b)$ in a graph $G = (V, E)$ via dynamic program

can be done in $\mathcal{O}\left(2^k \cdot k \cdot W \cdot |V(S_1)| \cdot |E||V|^2 \cdot \log\left(\frac{2|E|}{|V|}\right)\right)$

adapted from [Björklund et al., ’12]

k := number of sets $S_1, \ldots, S_k$

k := number of popular faces



Randomized FPT Runtime

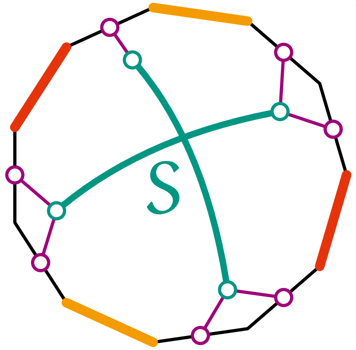
Computing $T_b(\mathcal{S}, m', b)$ in a graph $G = (V, E)$ via dynamic program

can be done in $\mathcal{O}\left(2^k \cdot k \cdot W \cdot |V(S_1)| \cdot |E||V|^2 \cdot \log\left(\frac{2|E|}{|V|}\right)\right)$

adapted from [Björklund et al., ’12]

Error: $1 - \frac{1}{n^W}$

k := number of sets $S_1, \dots, S_k$



k := number of popular faces

Randomized FPT Runtime

Computing $T_b(\mathcal{S}, m', b)$ in a graph $G = (V, E)$ via dynamic program

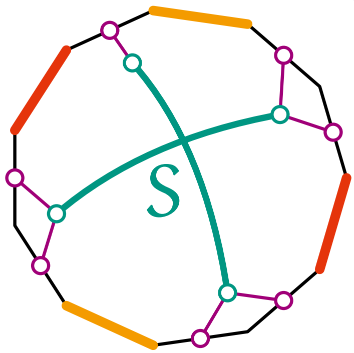
can be done in $\mathcal{O}\left(2^k \cdot k \cdot W \cdot |V(S_1)| \cdot |E||V|^2 \cdot \log\left(\frac{2|E|}{|V|}\right)\right)$

adapted from [Björklund et al., ’12]

Error: $1 - \frac{1}{n^W}$

#Vertices in smallest resolution set

k := number of sets $S_1, \ldots, S_k$



k := number of popular faces

Randomized FPT Runtime

Computing $T_b(\mathcal{S}, m', b)$ in a graph $G = (V, E)$ via dynamic program

can be done in $\mathcal{O}\left(2^k \cdot k \cdot W \cdot |V(S_1)| \cdot |E||V|^2 \cdot \log\left(\frac{2|E|}{|V|}\right)\right)$

adapted from [Björklund et al., ’12]

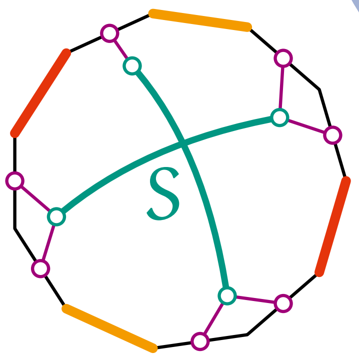
k := number of sets $S_1, \ldots, S_k$

Error: $1 - \frac{1}{n^W}$

#Vertices in smallest resolution set

Resolving all popular faces can be done in $\mathcal{O}\left(2^k \text{poly}(n)\right)$ with one sided Error

k := number of popular faces



Randomized FPT Runtime

Computing $T_b(\mathcal{S}, m', b)$ in a graph $G = (V, E)$ via dynamic program

can be done in $\mathcal{O}\left(2^k \cdot k \cdot W \cdot |V(S_1)| \cdot |E||V|^2 \cdot \log\left(\frac{2|E|}{|V|}\right)\right)$

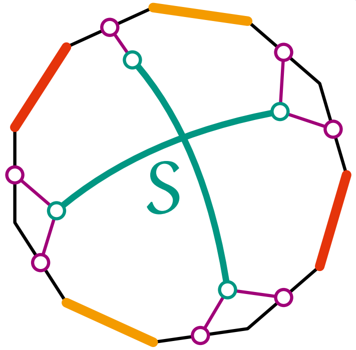
adapted from [Björklund et al., ’12]

k := number of sets $S_1, \ldots, S_k$

Error: $1 - \frac{1}{n^W}$

#Vertices in smallest resolution set

Resolving all popular faces can be done in $\mathcal{O}\left(2^k \text{poly}(n)\right)$ with one sided Error



k := number of popular faces

Randomized FPT

Outline

Nonograms

How to remove popular faces

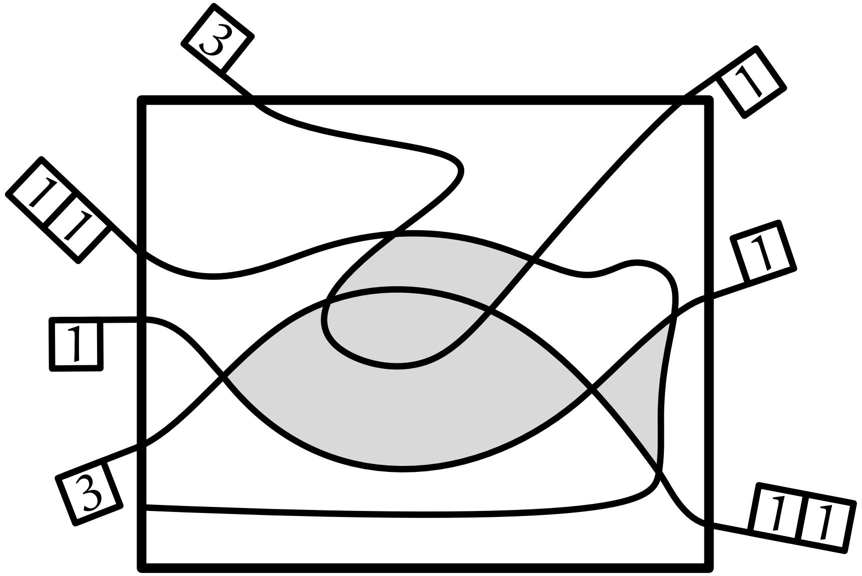
Resolution with one curve is NP-complete...

...but we can do it randomized in FPT time

Summary

Wrap-Up

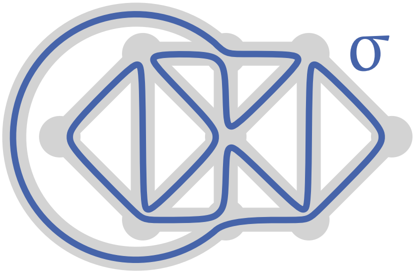
Advanced nonograms can turned into basic nonograms by adding **additional resolution curves**.

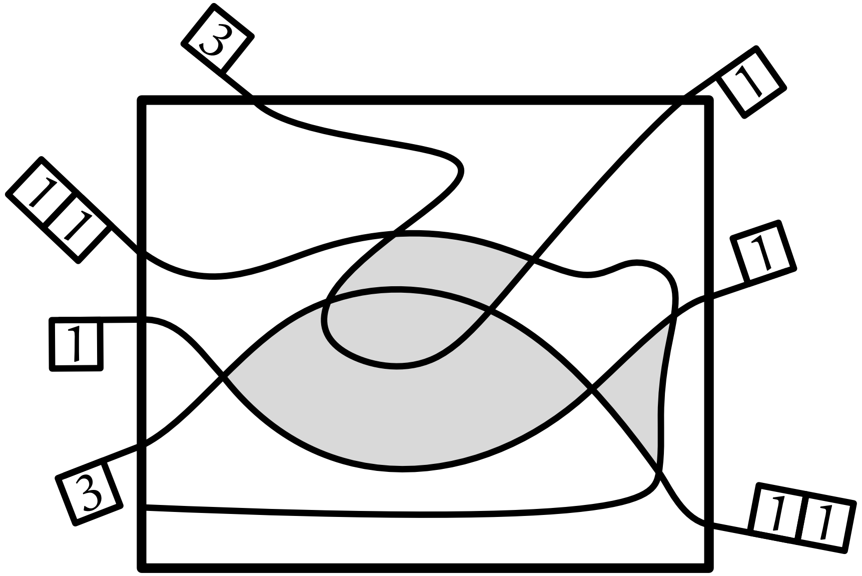


Wrap-Up

Advanced nonograms can turned into basic nonograms by adding **additional resolution curves**.

Deciding if **one curve** is sufficient is **NP-complete**...

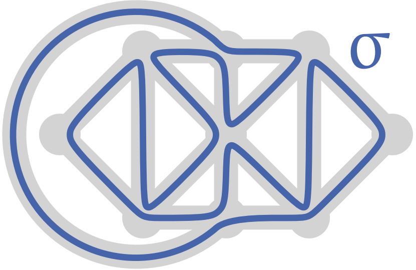


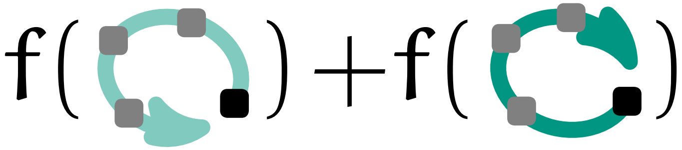


Wrap-Up

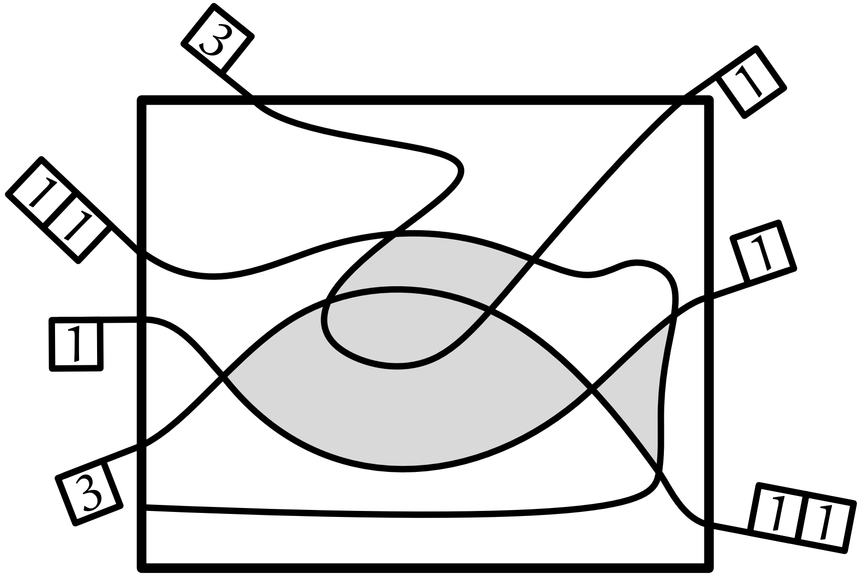
Advanced nonograms can turned into basic nonograms by adding **additional resolution curves**.

Deciding if **one curve** is sufficient is **NP-complete**...





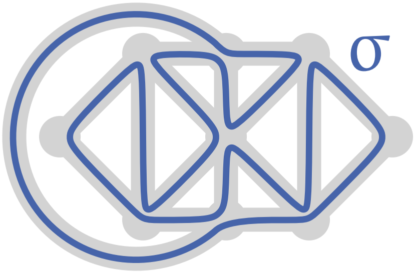
...but possible in **randomized FPT** with exponentially small **one-sided** error.

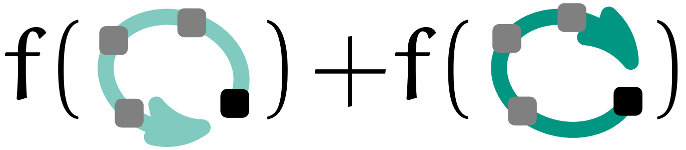


Wrap-Up

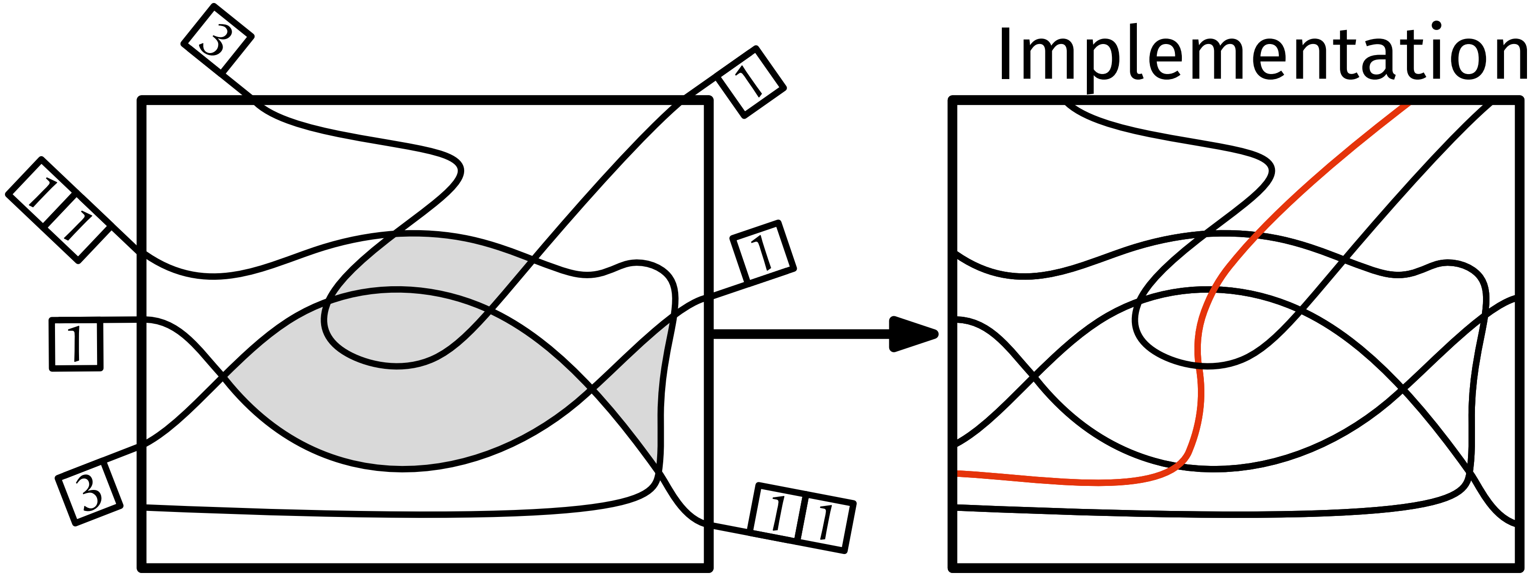
Advanced nonograms can turned into basic nonograms by adding **additional resolution curves**.

Deciding if **one curve** is sufficient is **NP-complete**...





...but possible in **randomized FPT** with exponentially small **one-sided** error.

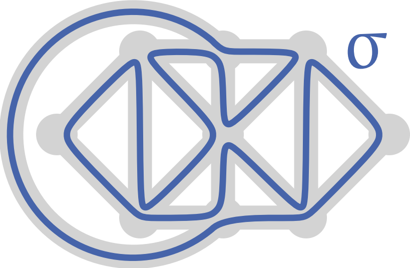


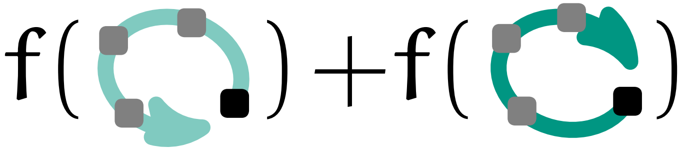
Implementation

Wrap-Up

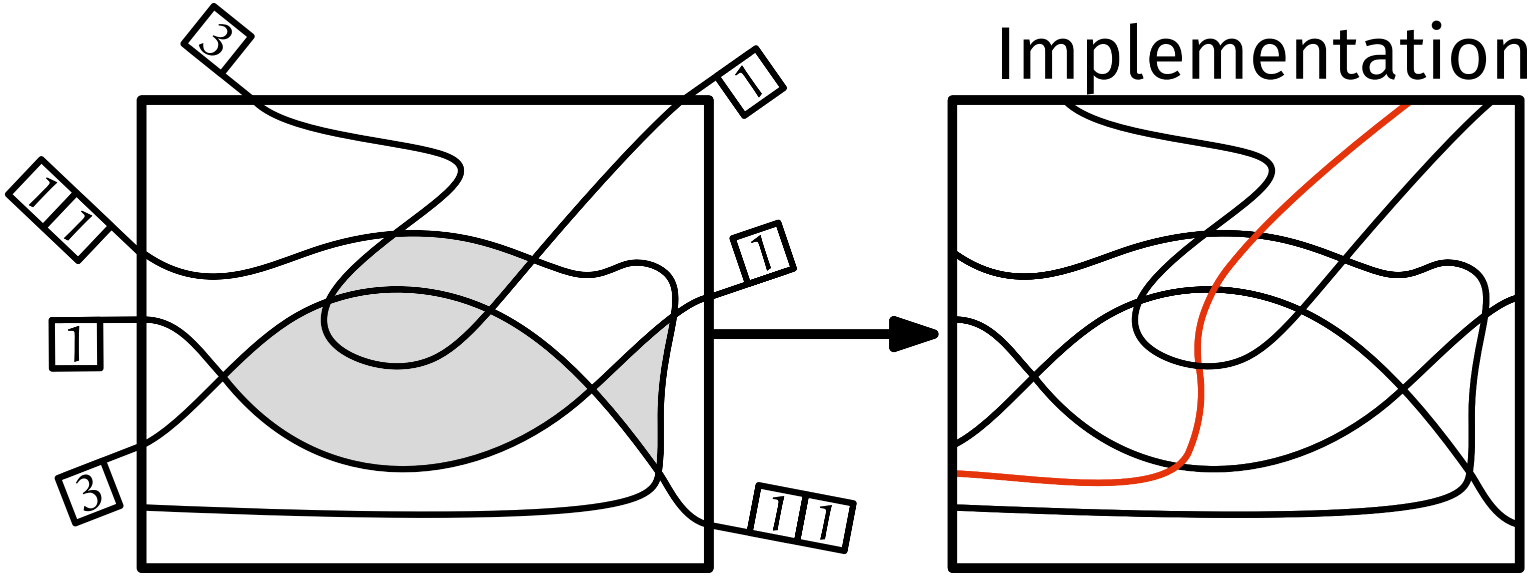
Advanced nonograms can turned into basic nonograms by adding **additional resolution curves**.

Deciding if **one curve** is sufficient is **NP-complete**...





...but possible in **randomized FPT** with exponentially small **one-sided** error.



Open questions:

- More curves?
- Eliminating the error?

Intuition on the One-Sided Error

$$\begin{array}{r} [11010001] \\ +[11010001] \\ \hline [00000000] \end{array}$$

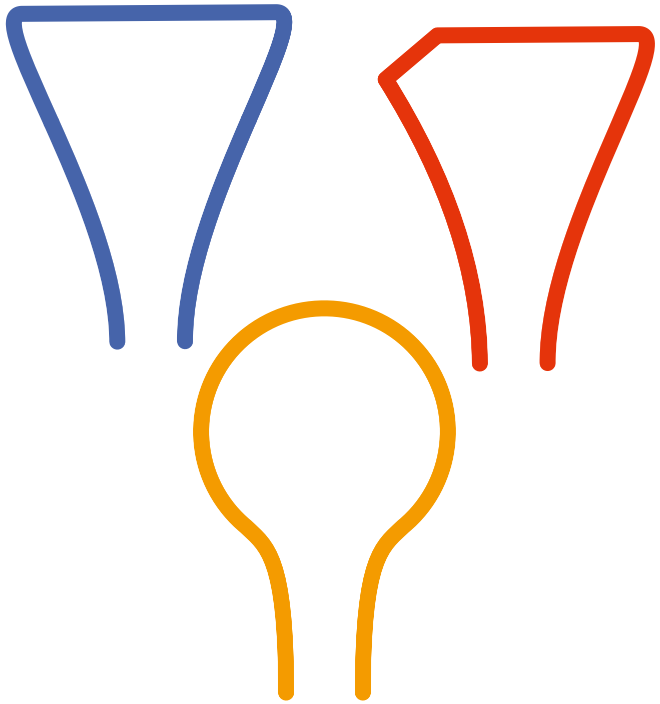
Intuition on the One-Sided Error

$$\begin{array}{r} [11010001] \\ +[11010001] \\ \hline [00000000] \end{array}$$

$$\begin{array}{r} [11010001] \\ +[10010101] \\ \hline [01000100] \end{array} \qquad \begin{array}{r} [11010001] \\ +[01000100] \\ \hline [10010101] \end{array} \qquad \begin{array}{r} [10010101] \\ +[01000100] \\ \hline [11010001] \end{array} \qquad \begin{array}{r} [10010101] \\ +[01000100] \\ +[11010001] \\ \hline [00000000] \end{array}$$

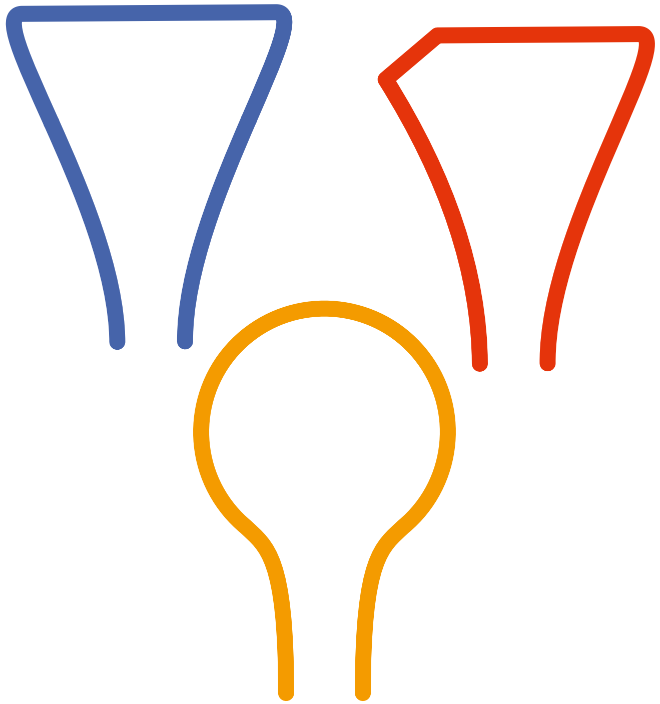
The Reduction

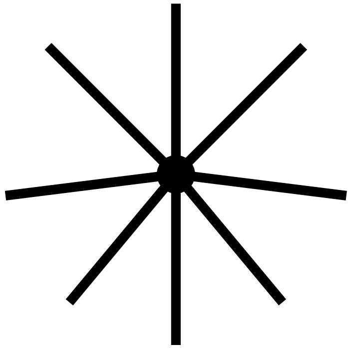
The building block curves



The Reduction

The building block curves

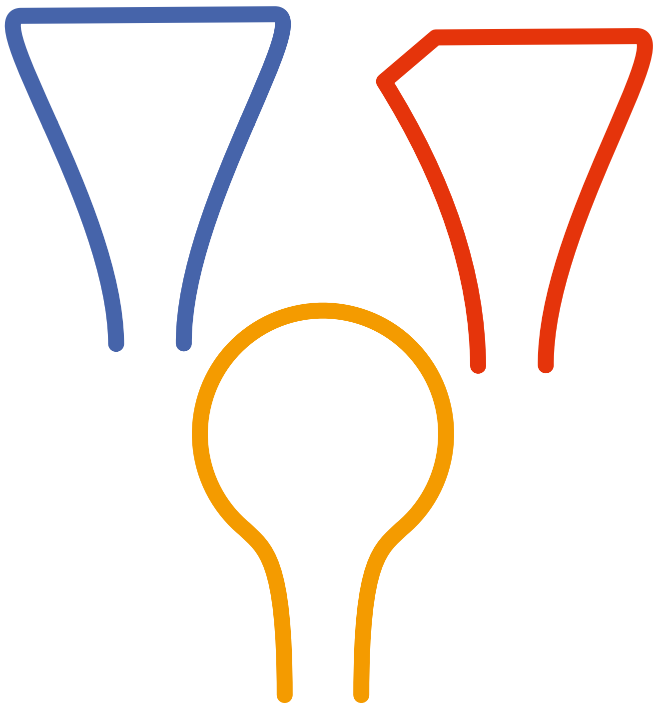


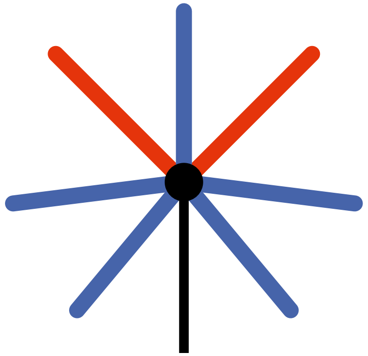


Vertex v

The Reduction

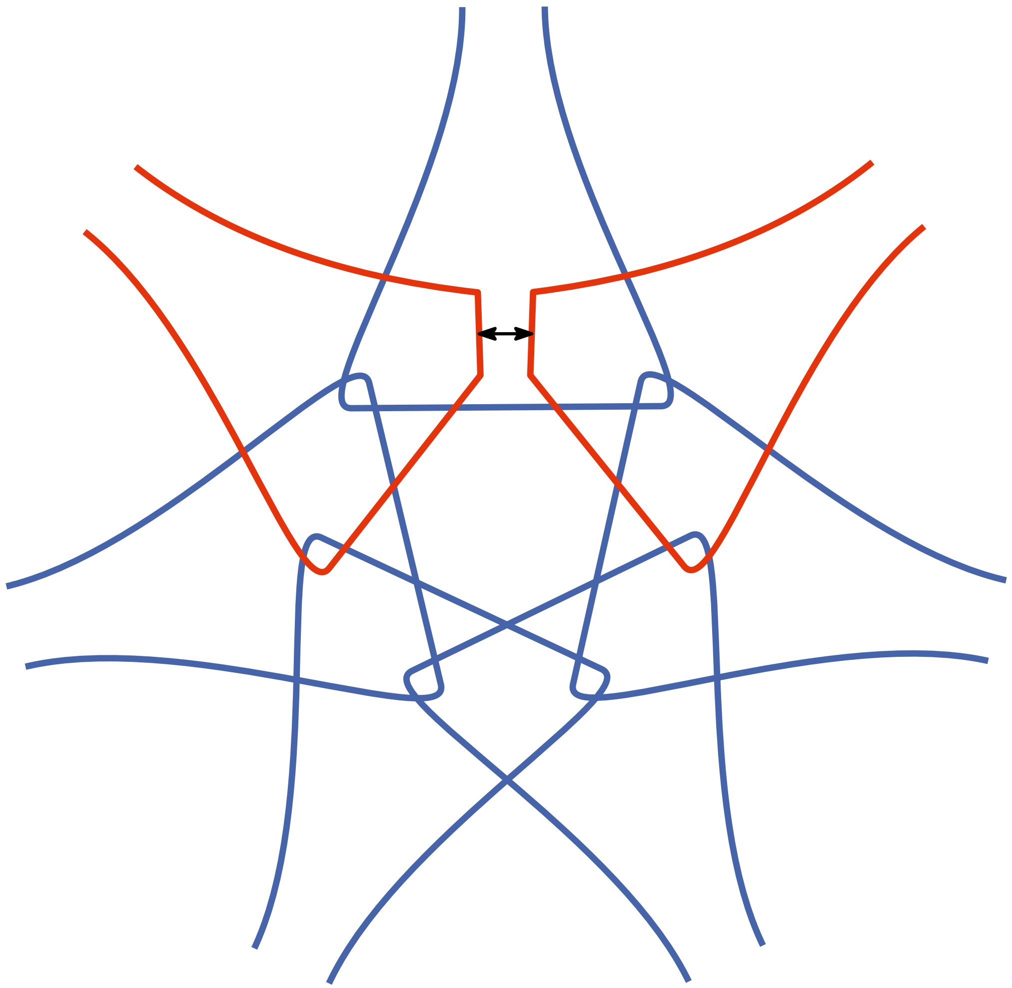
The building block curves





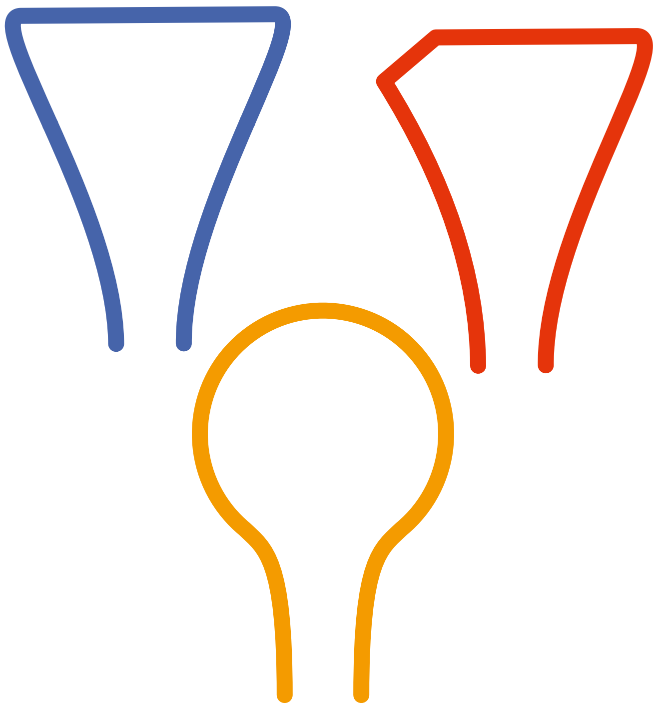
Vertex v

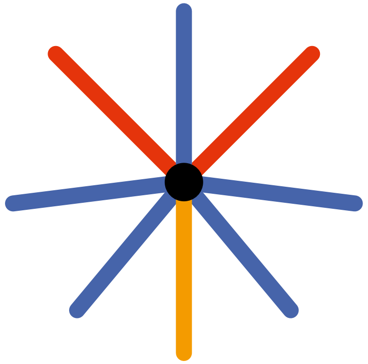
Vertex gadget $\mathcal{G}(v)$



The Reduction

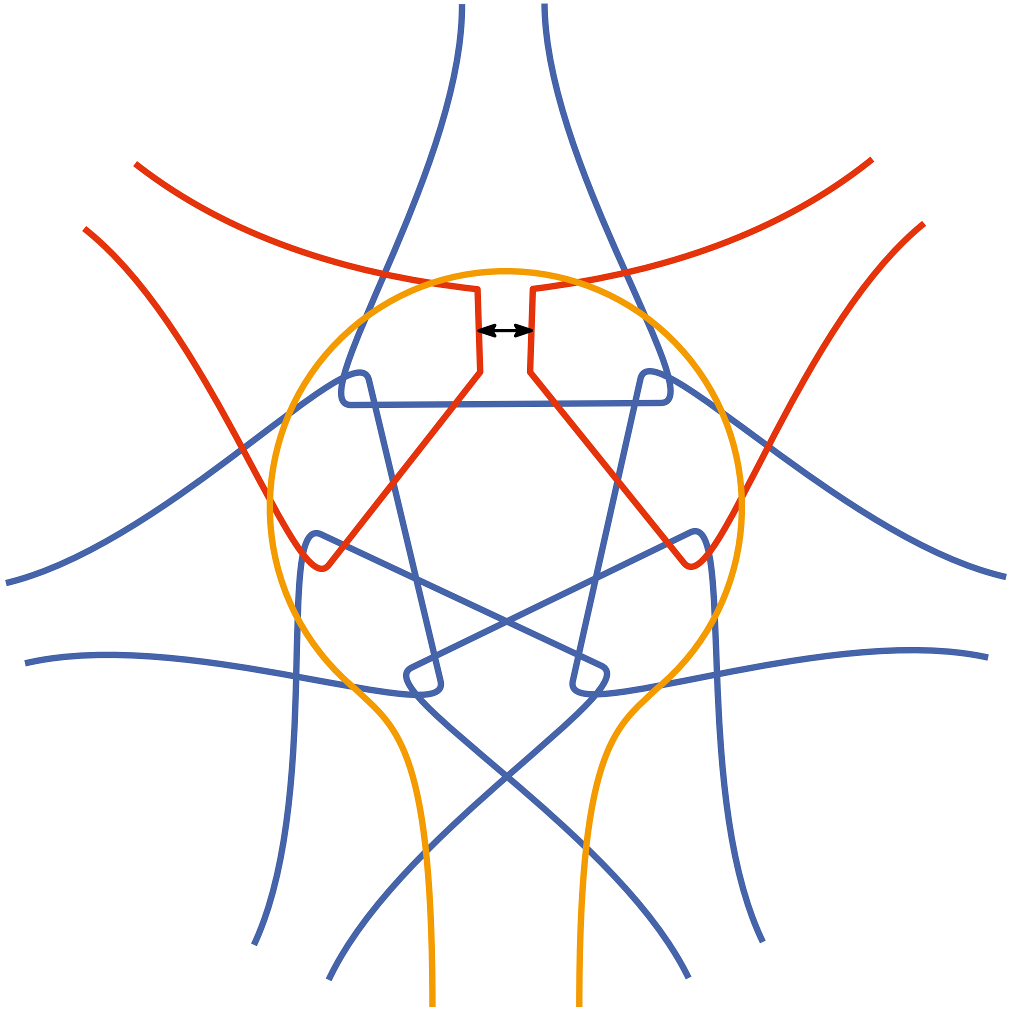
The building block curves





Vertex v

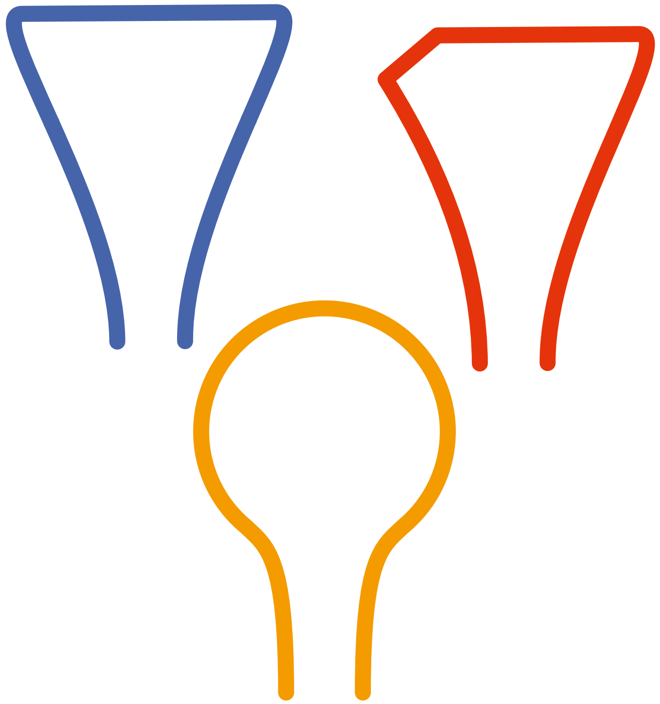
Vertex gadget $\mathcal{G}(v)$

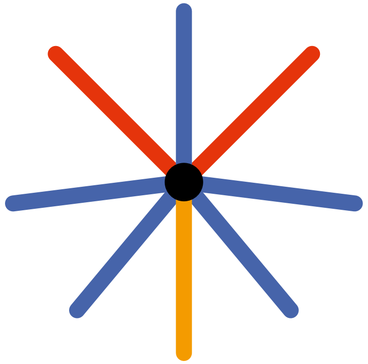


- Number of “openings” = $\deg(v)$

The Reduction

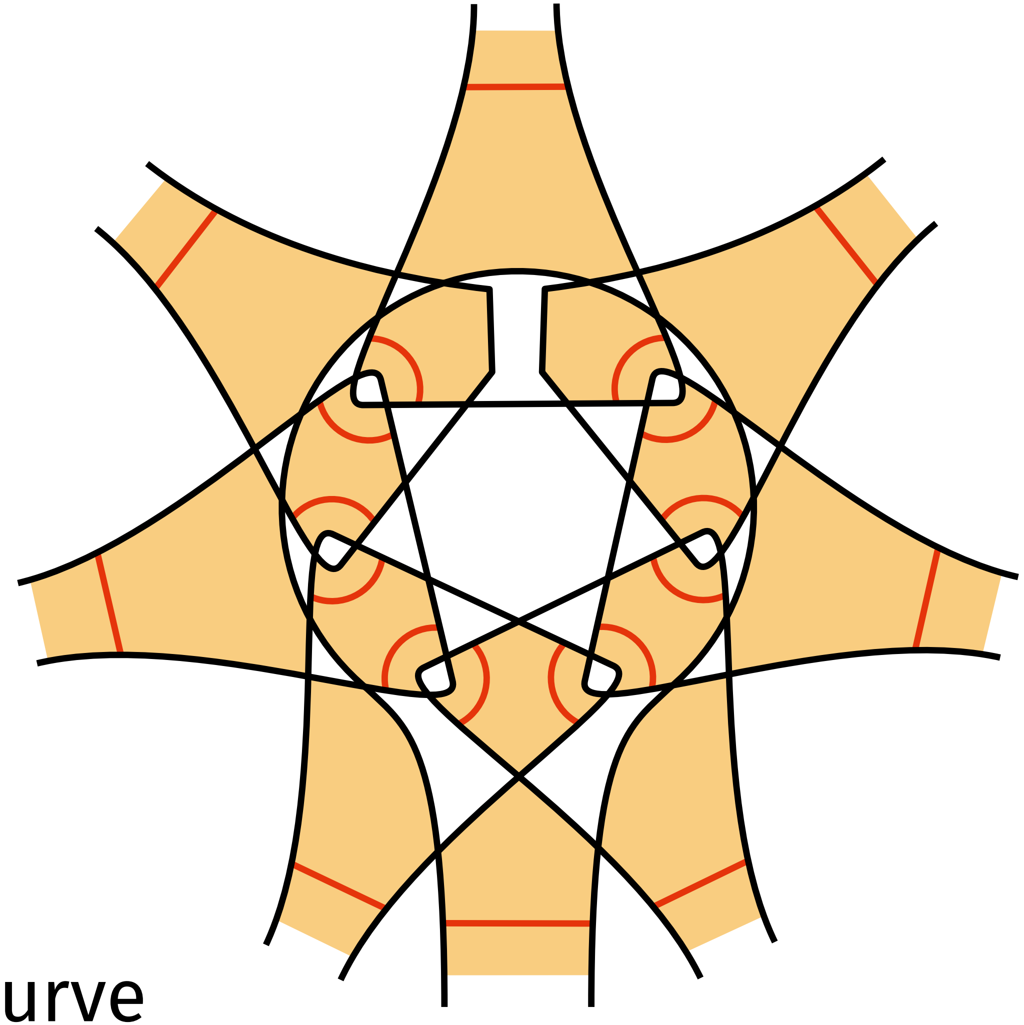
The building block curves





Vertex v

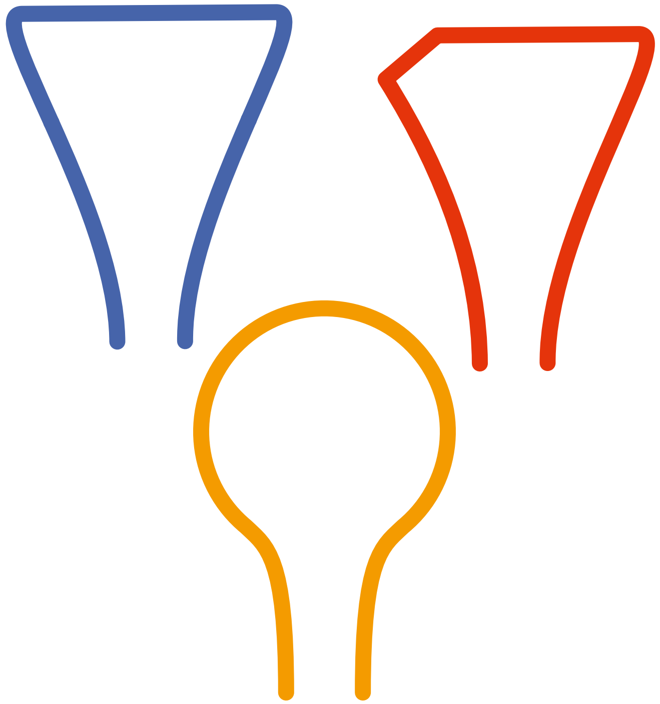
Vertex gadget $\mathcal{G}(v)$

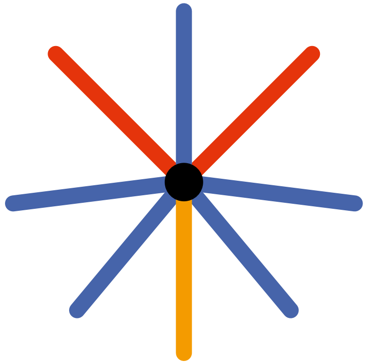


- Number of “openings” = $\deg(v)$
- Popular faces force resolution curve c

The Reduction

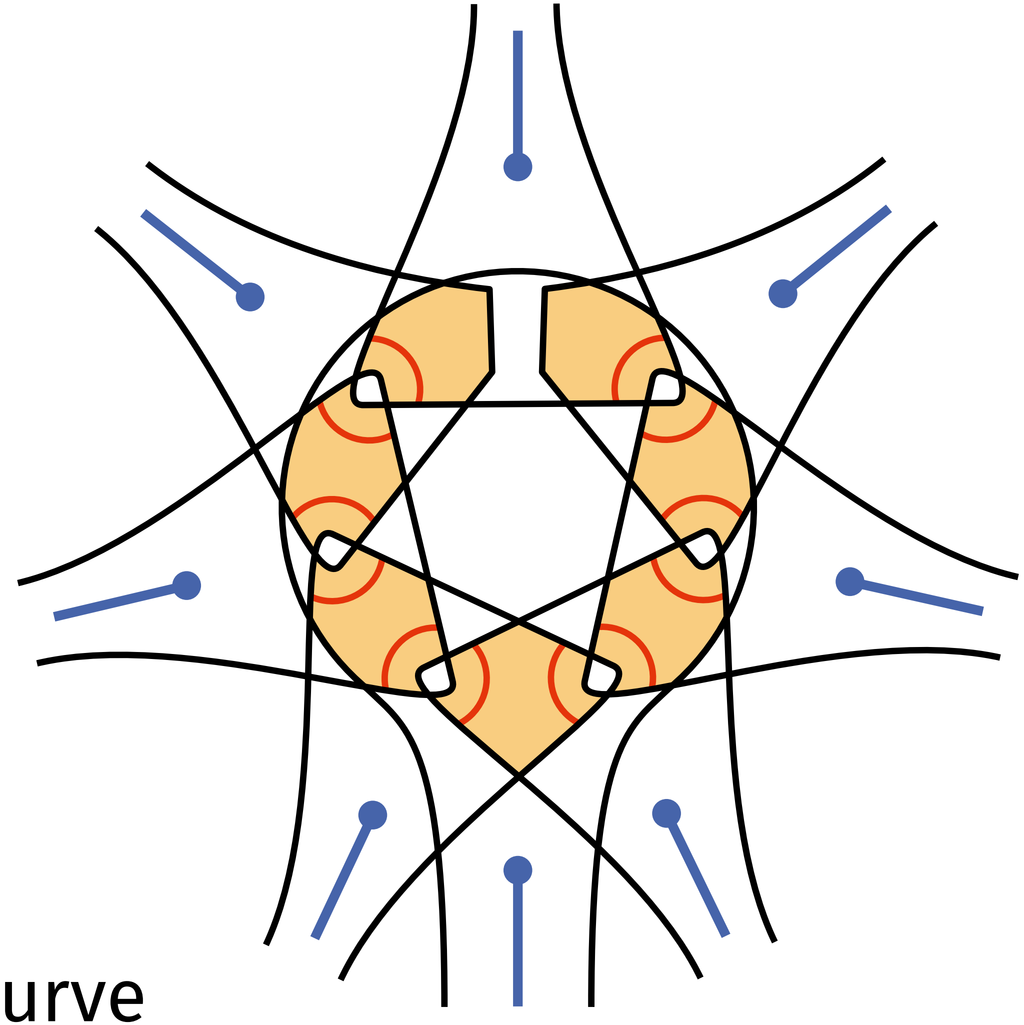
The building block curves





Vertex v

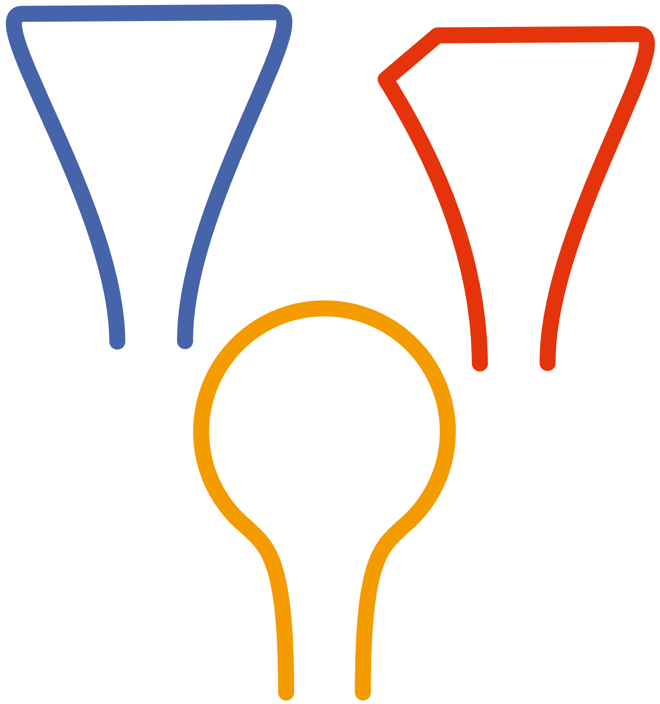
Vertex gadget $\mathcal{G}(v)$

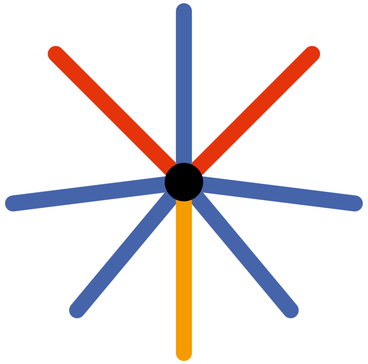


- Number of “openings” = $\deg(v)$
- Popular faces force resolution curve c

The Reduction

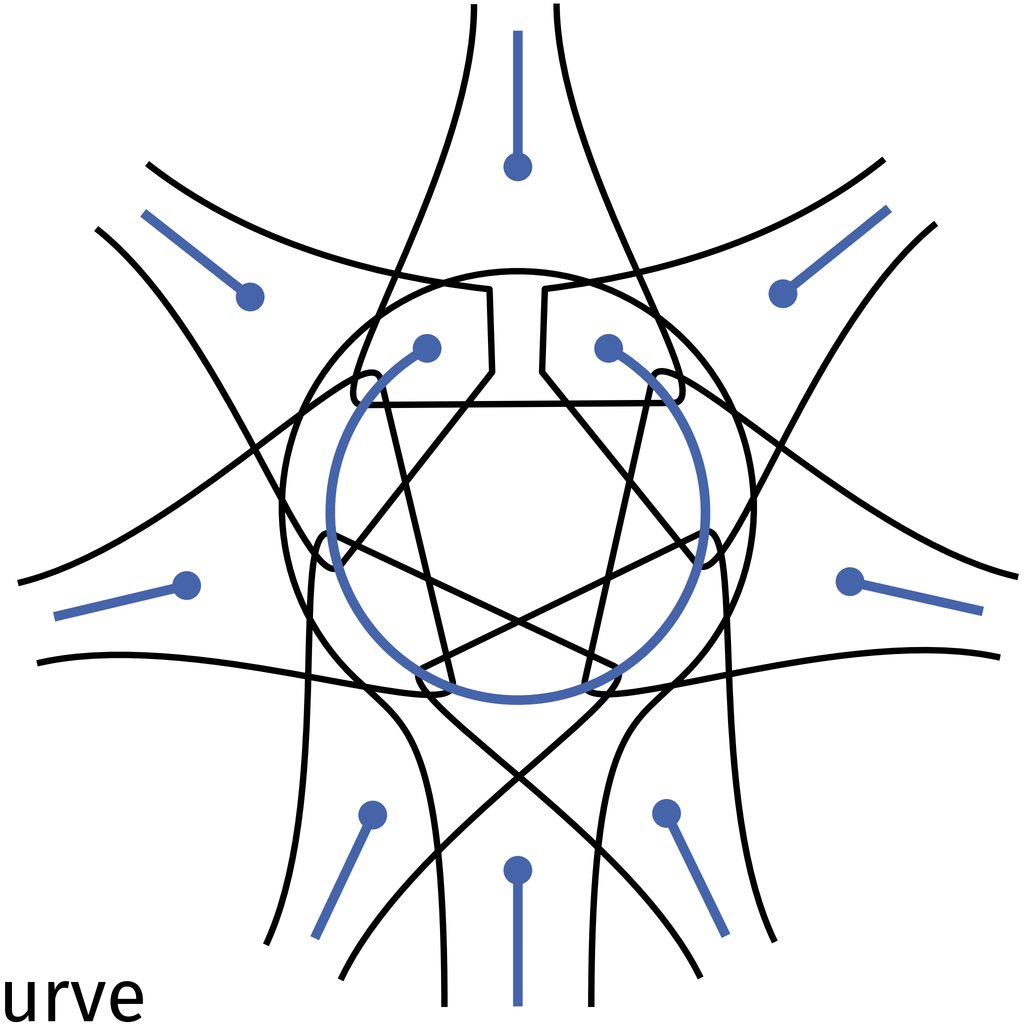
The building block curves





Vertex v

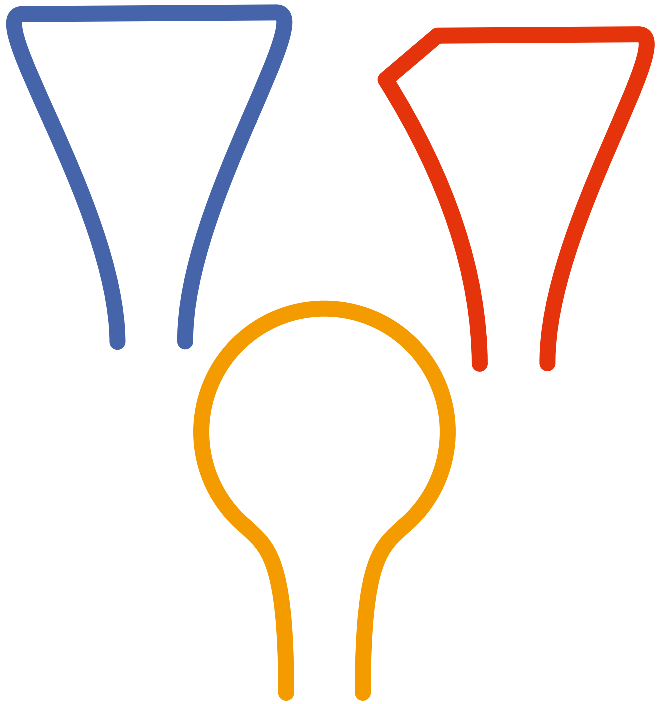
Vertex gadget $\mathcal{G}(v)$

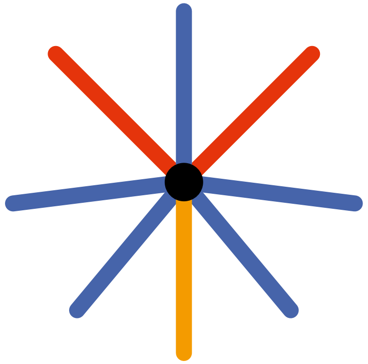


- Number of “openings” = $\deg(v)$
- Popular faces force resolution curve c

The Reduction

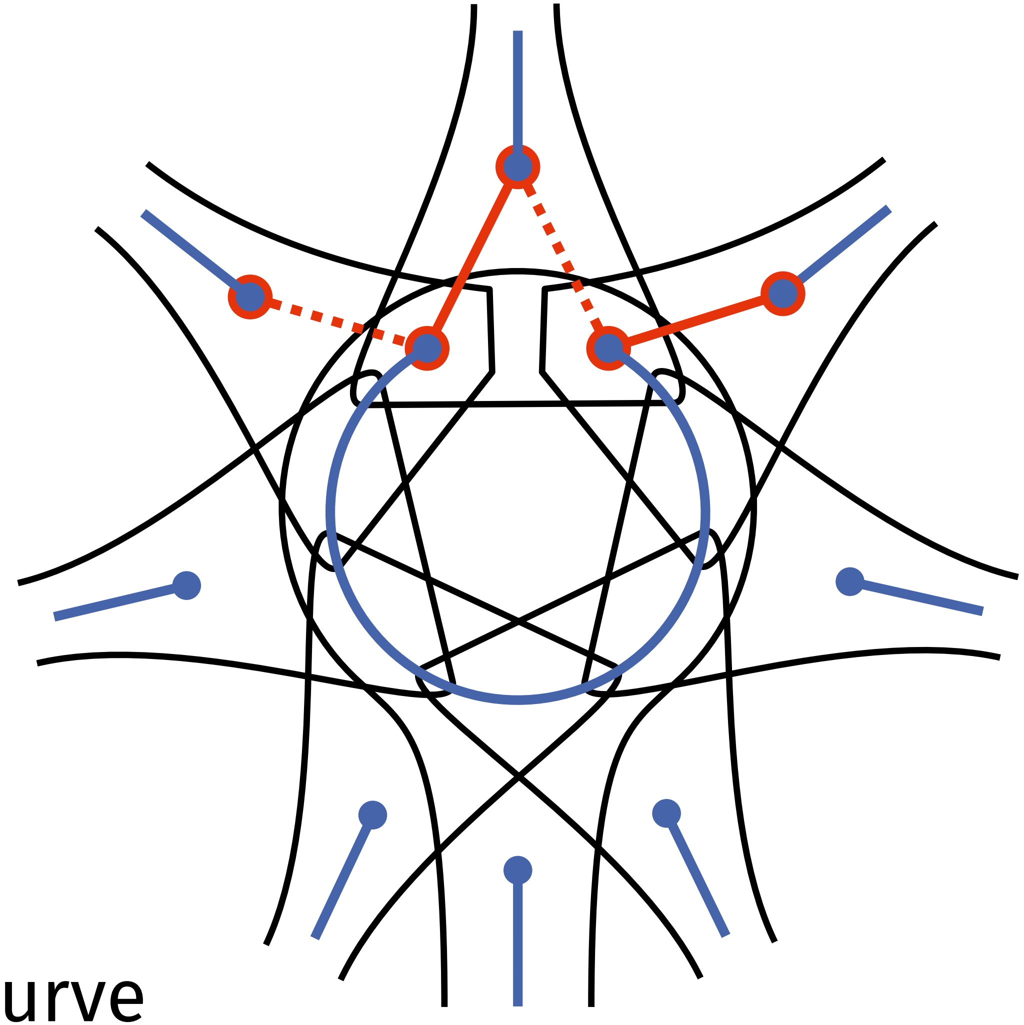
The building block curves





Vertex v

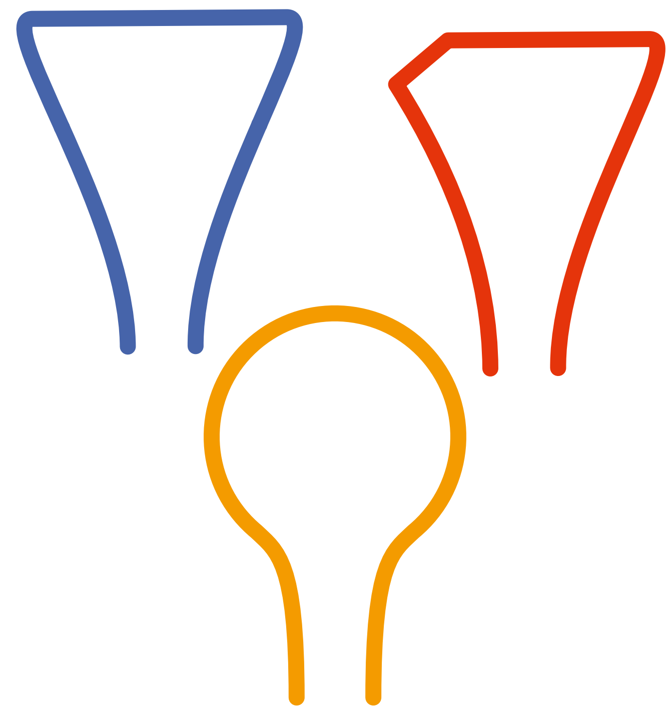
Vertex gadget $\mathcal{G}(v)$

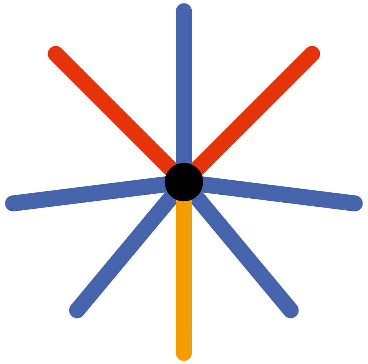


- Number of “openings” = $\deg(v)$
- Popular faces force resolution curve c
- Inner part forces one of two options

The Reduction

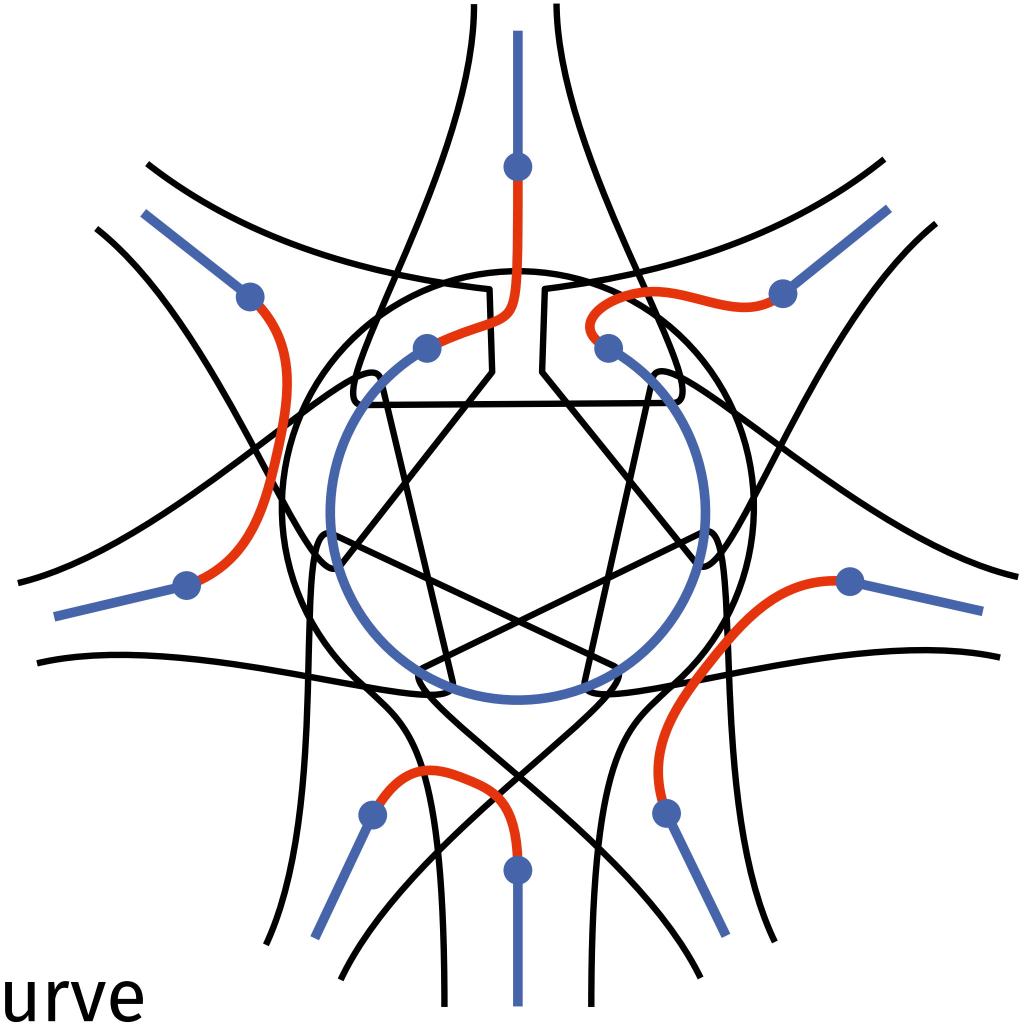
The building block curves





Vertex v

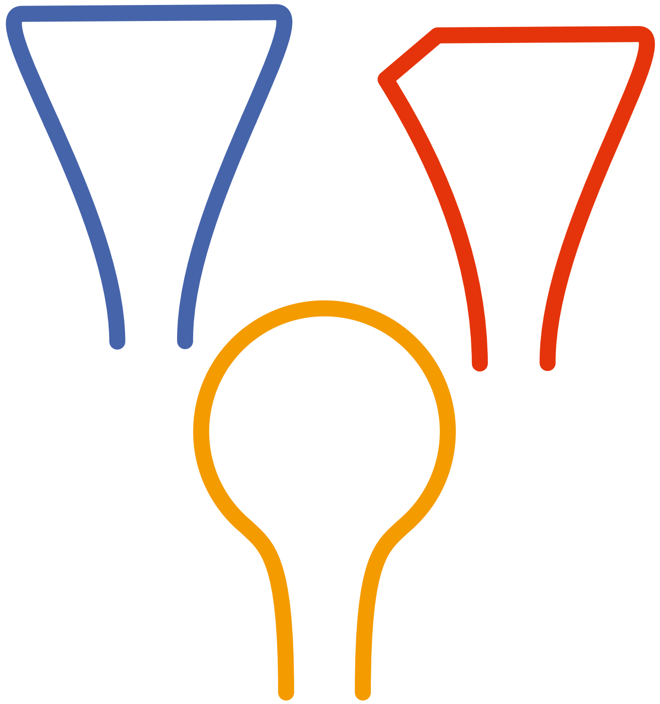
Vertex gadget $\mathcal{G}(v)$

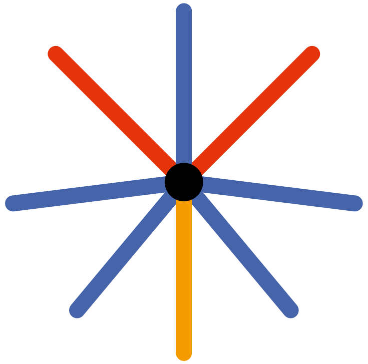


- Number of “openings” = $\deg(v)$
- Popular faces force resolution curve c
- Inner part forces one of two options

The Reduction

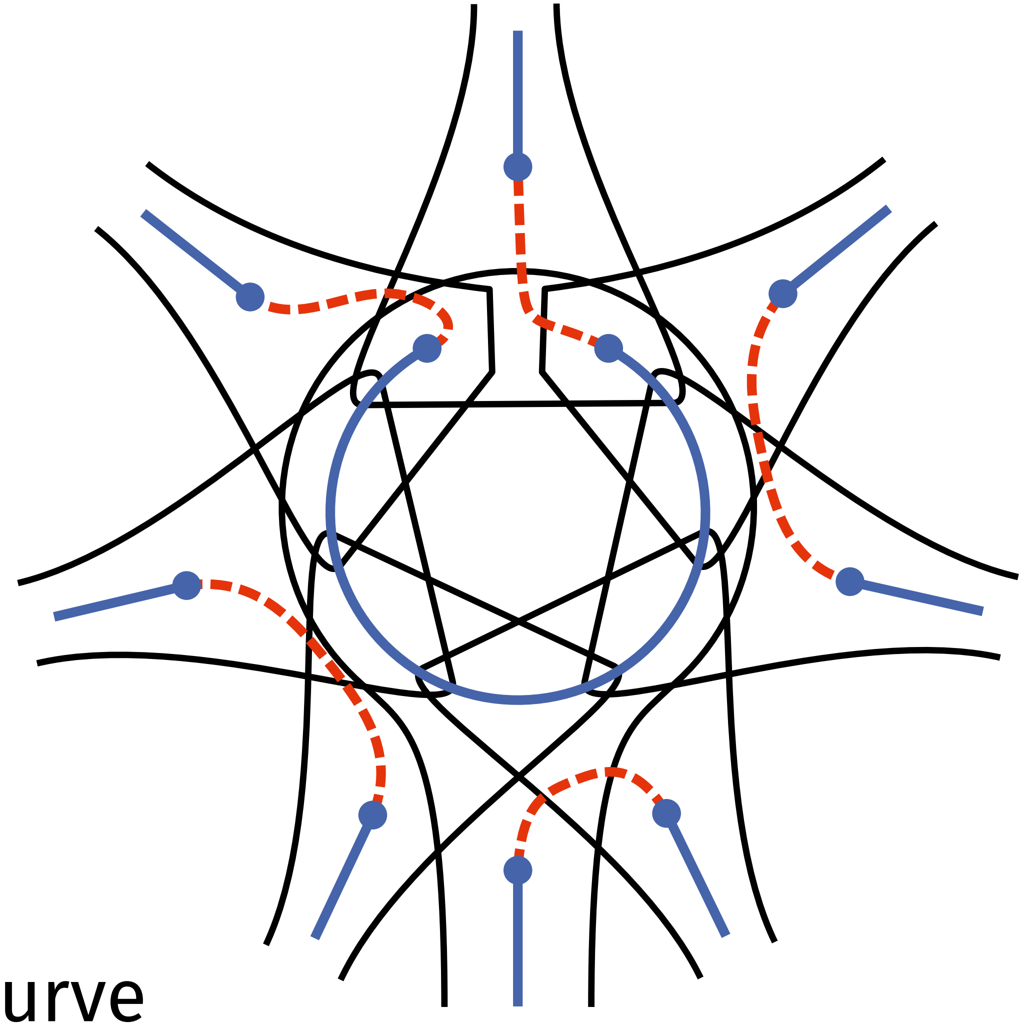
The building block curves





Vertex v

Vertex gadget $\mathcal{G}(v)$



- Number of “openings” = $\deg(v)$
- Popular faces force resolution curve c
- Inner part forces one of two options

The Edge Gadget