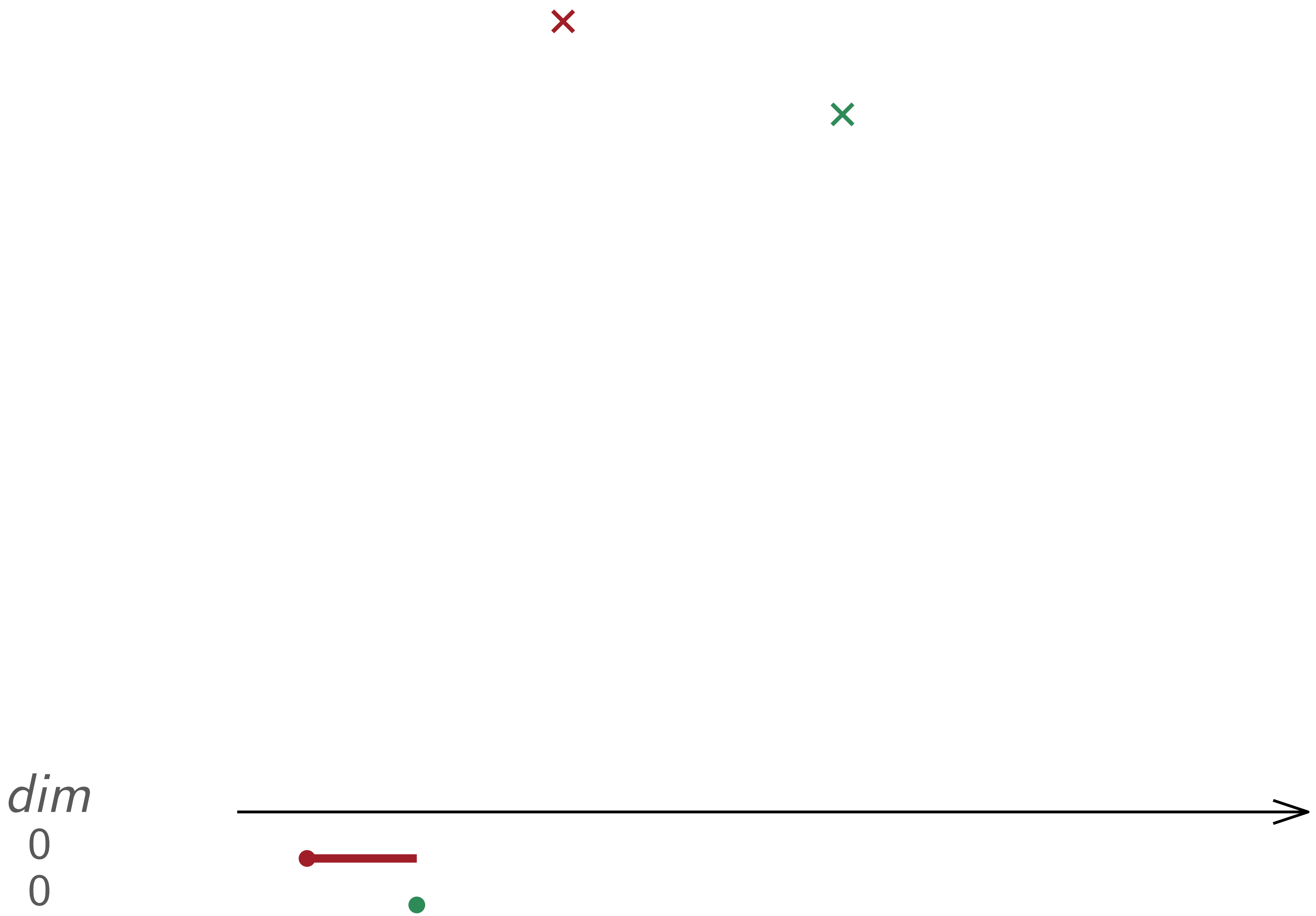
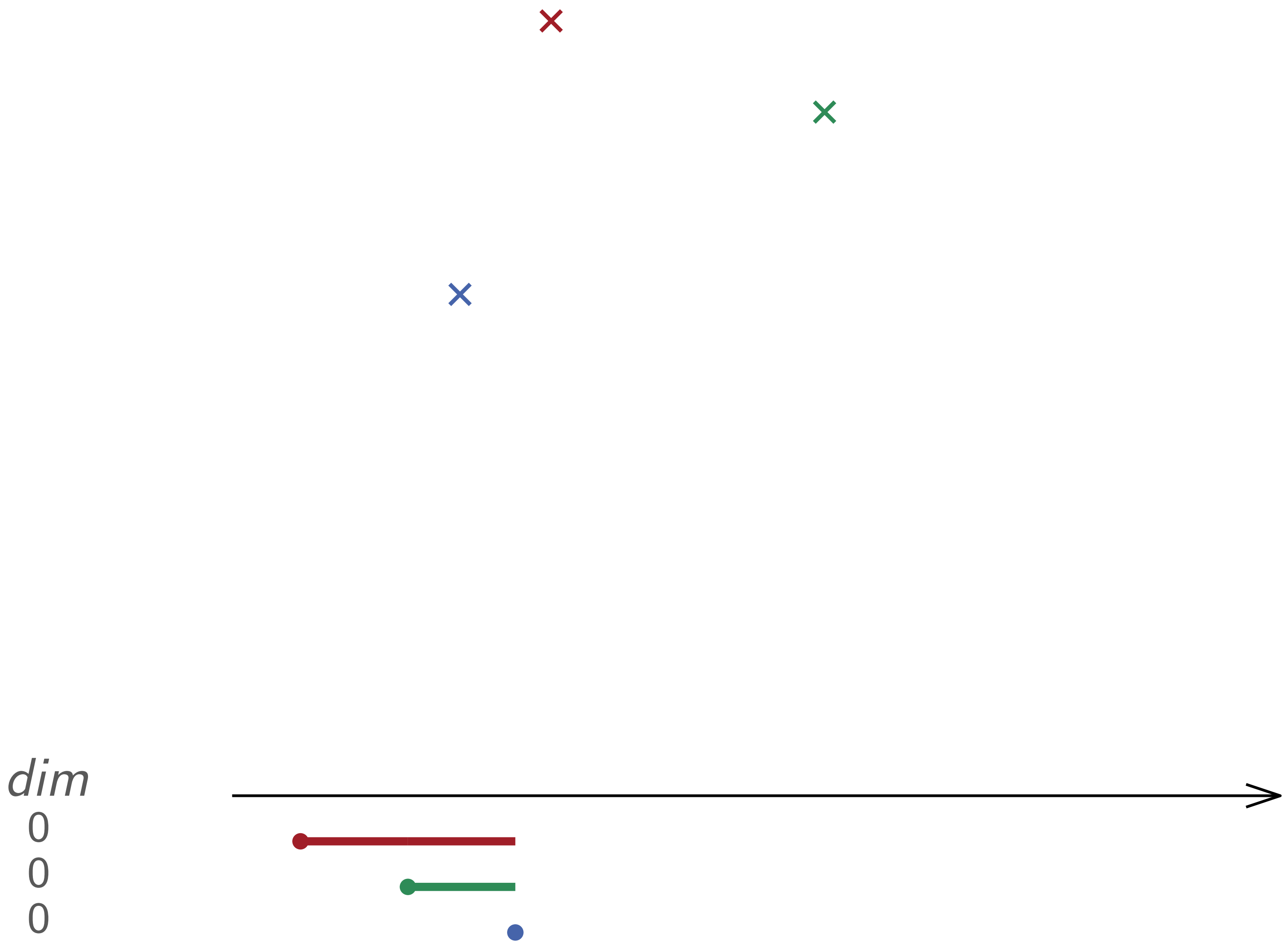
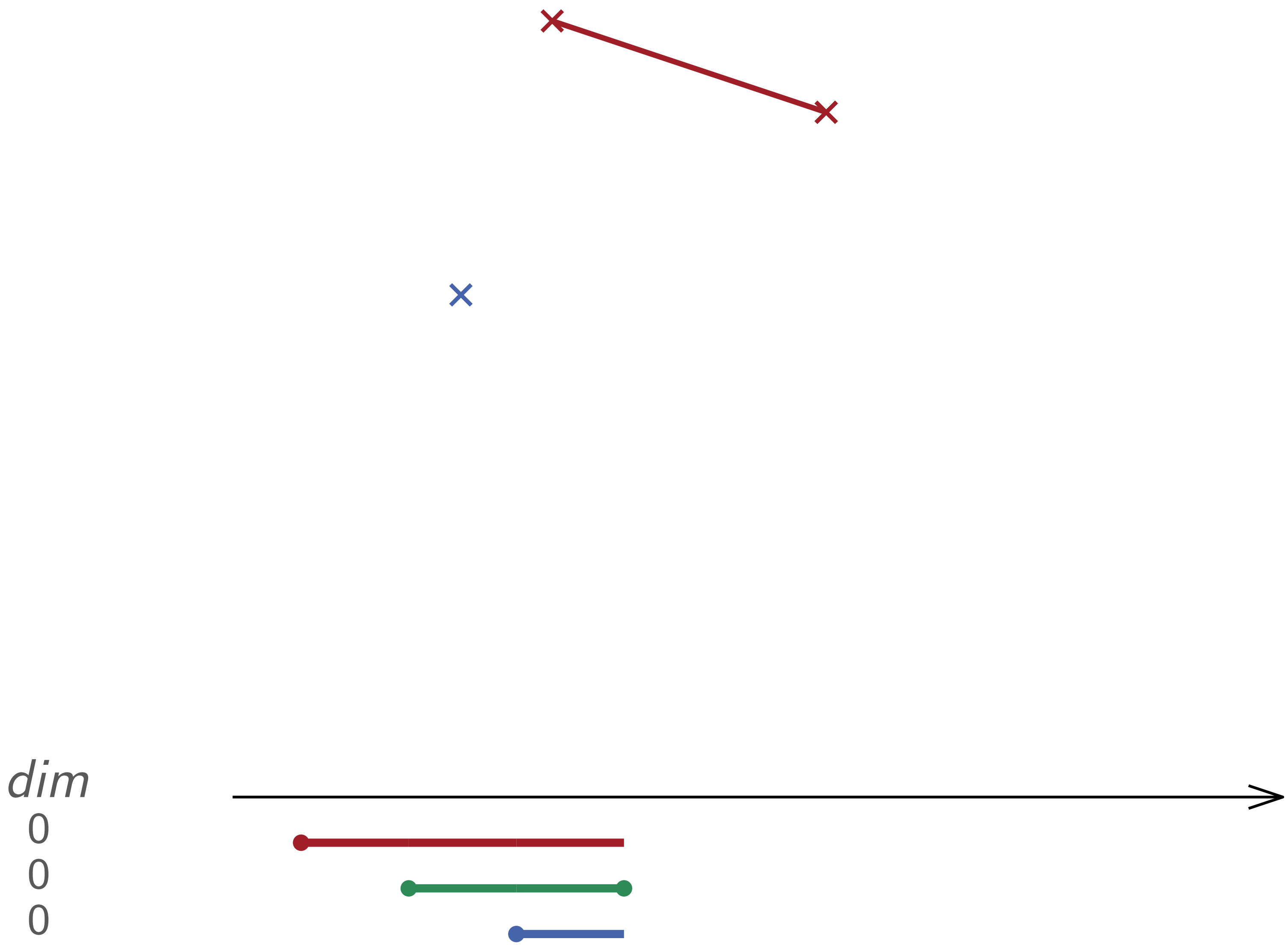
Barcodes of Towers and a Streaming Algorithm for Persistent Homology

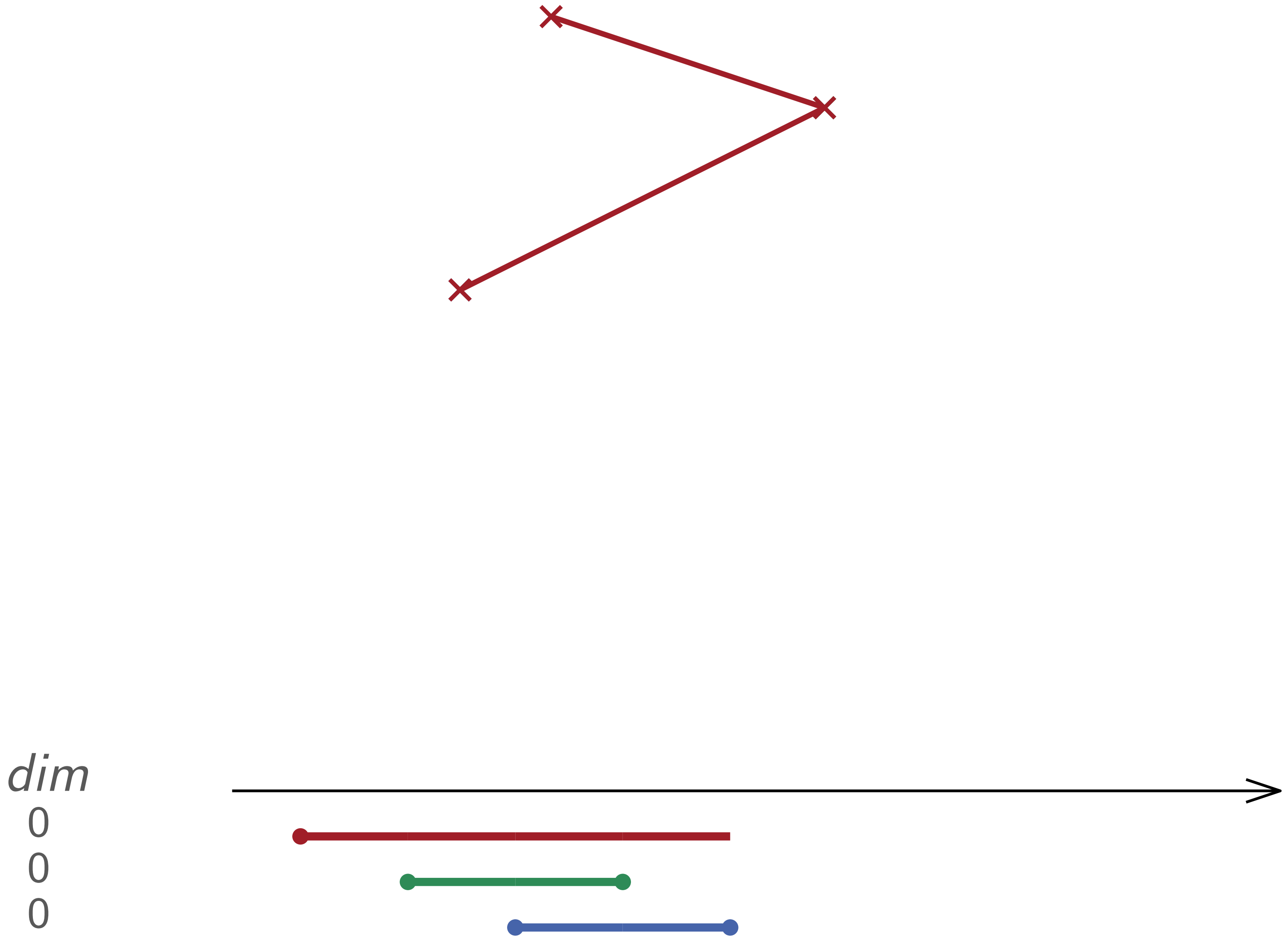
Hannah Schreiber
Joint with Michael Kerber
Gudhi Workshop, October 2016

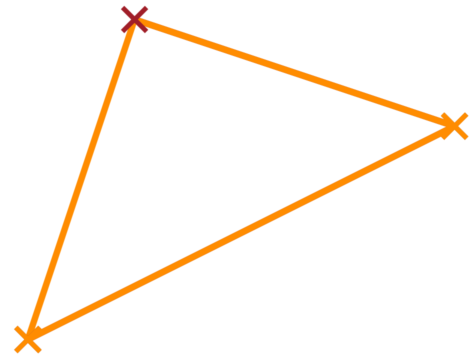
Persistent Homology

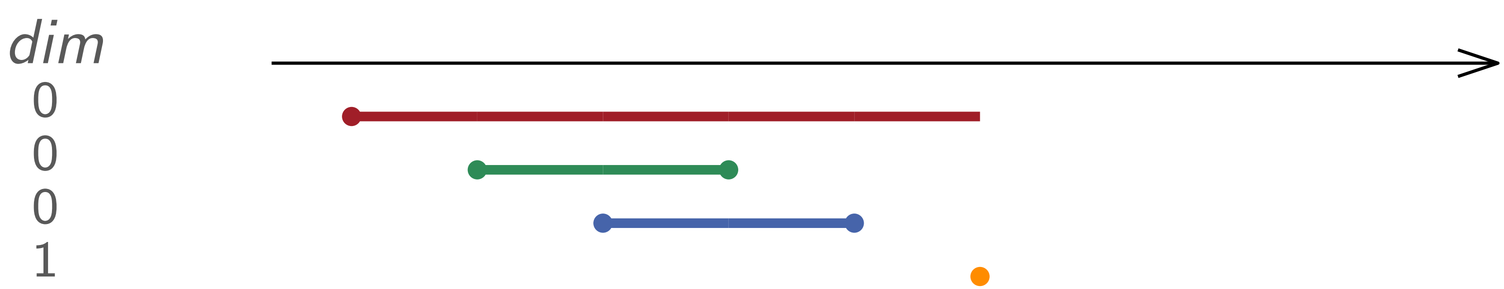

dim
0
0

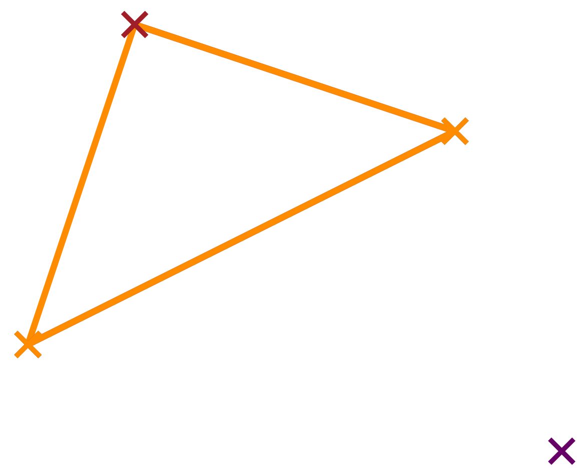

dim
0
0
0

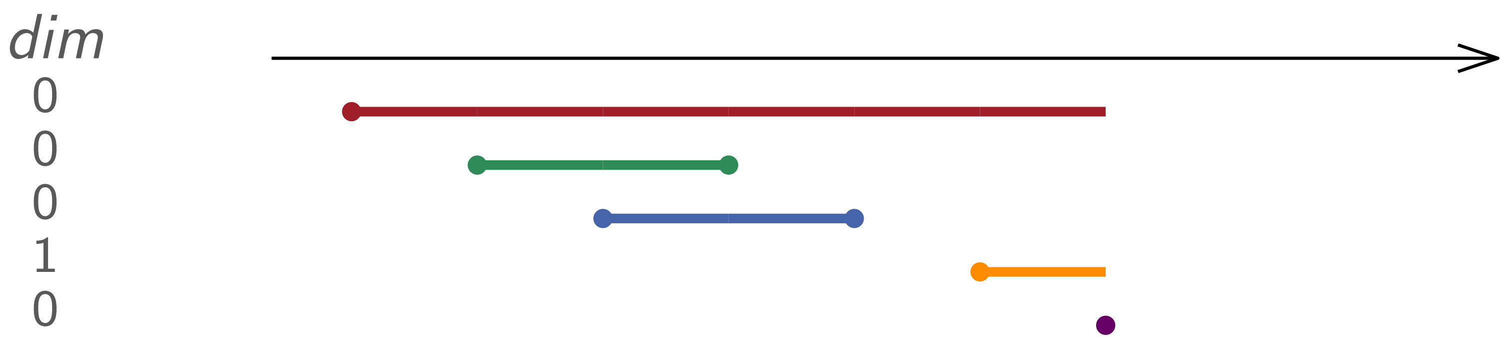

dim
0
0
0


dim
0
0
0

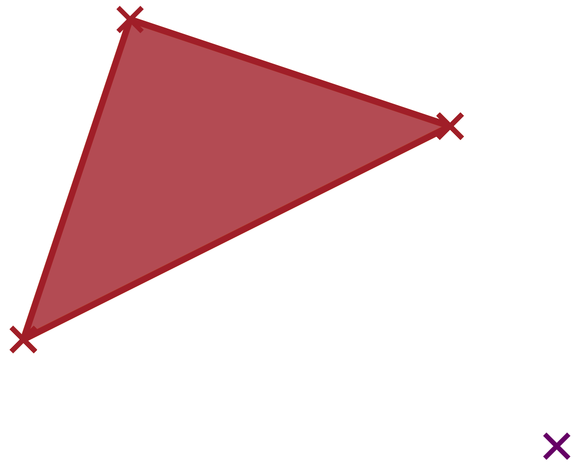


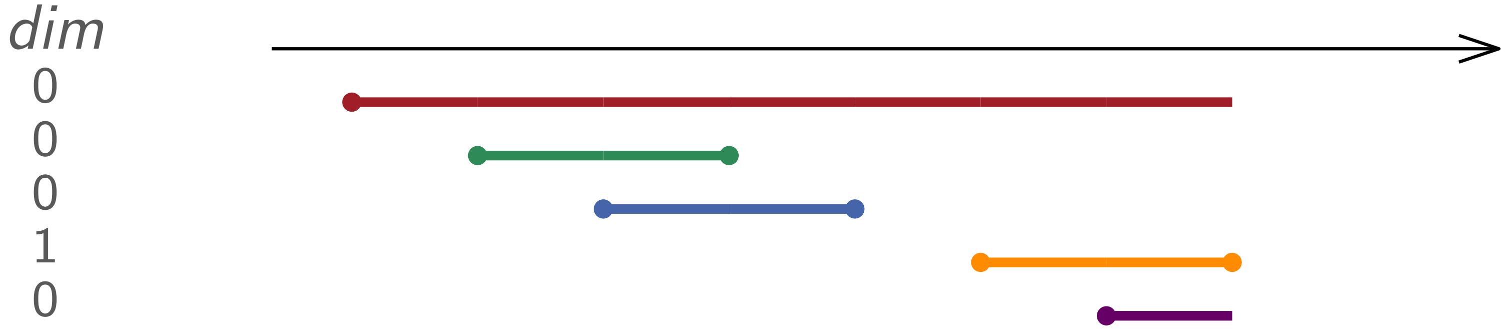

dim
0
0
0
1



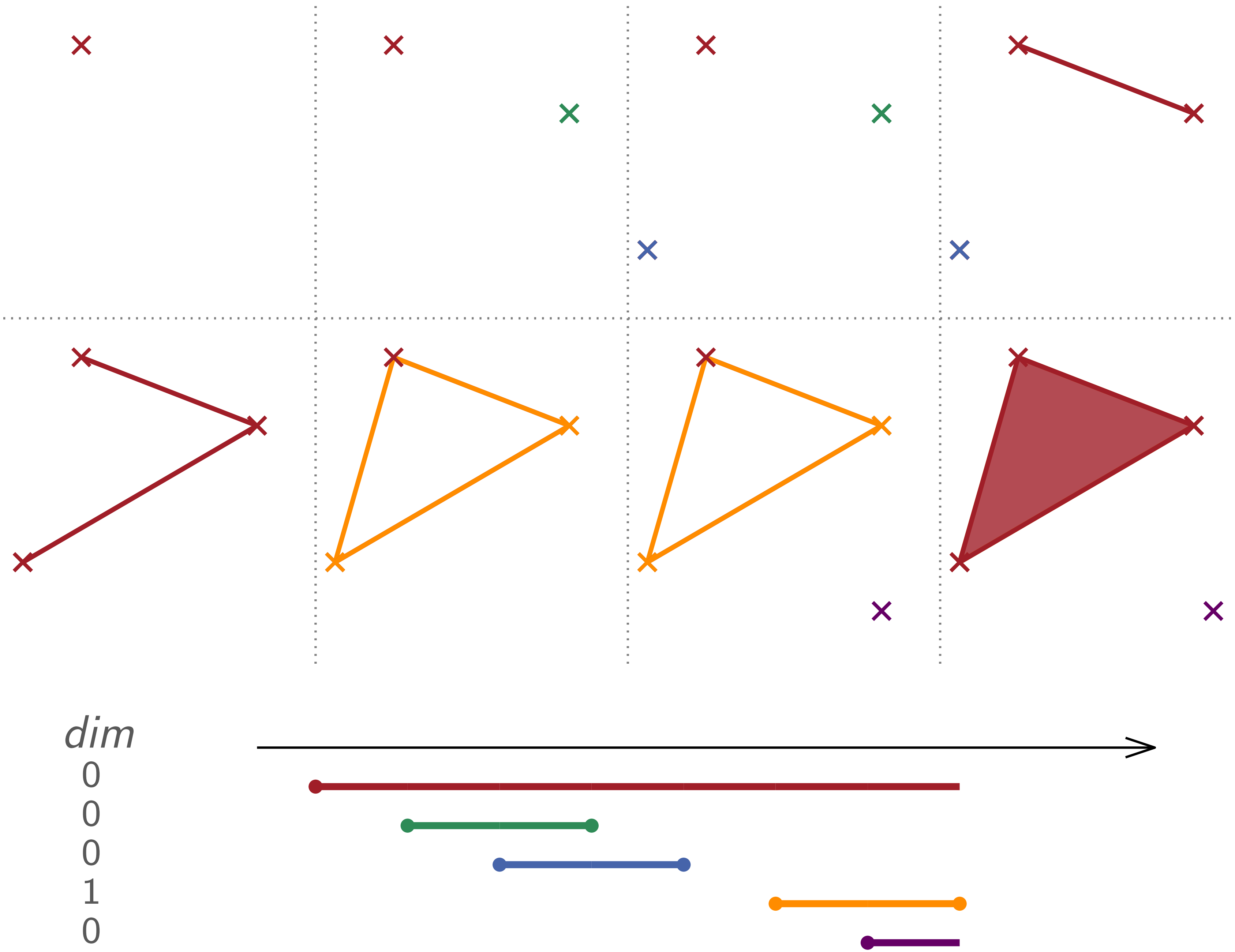

dim
0
0
0
1
0

Persistent Homology





Persistent Homology

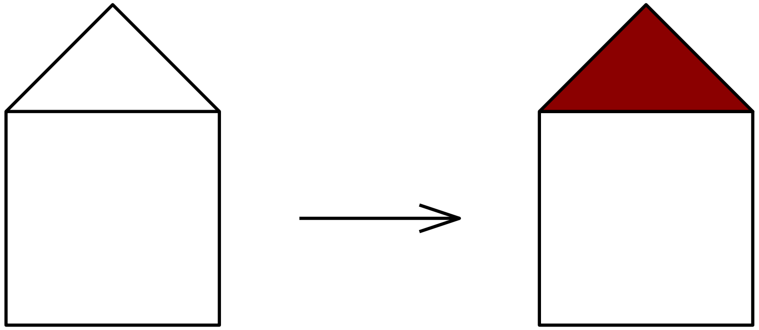


Filtration:

$$\mathbb{K}_0 \hookrightarrow \mathbb{K}_1 \hookrightarrow \ldots \hookrightarrow \mathbb{K}_n$$

Filtration:

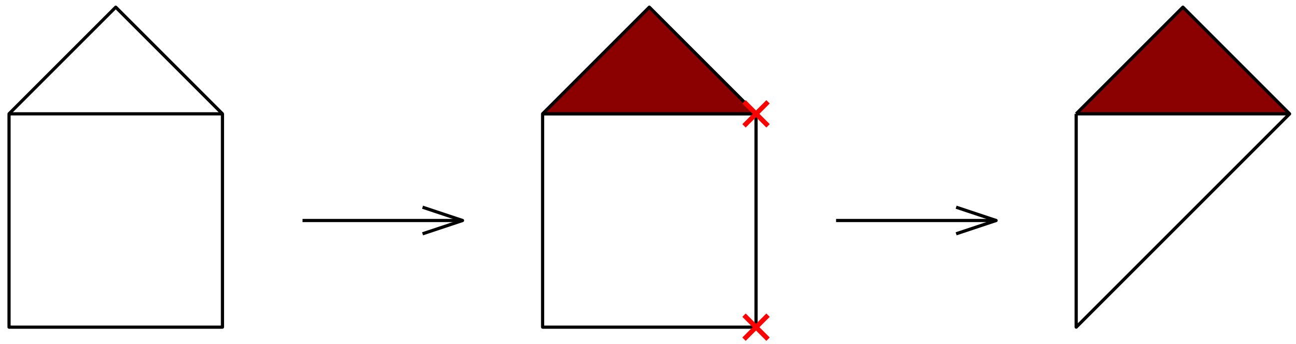
$$\mathbb{K}_0 \hookrightarrow \mathbb{K}_1 \hookrightarrow \ldots \hookrightarrow \mathbb{K}_n$$



Towers

Filtration:

$$\mathbb{K}_0 \hookrightarrow \mathbb{K}_1 \hookrightarrow \ldots \hookrightarrow \mathbb{K}_n$$



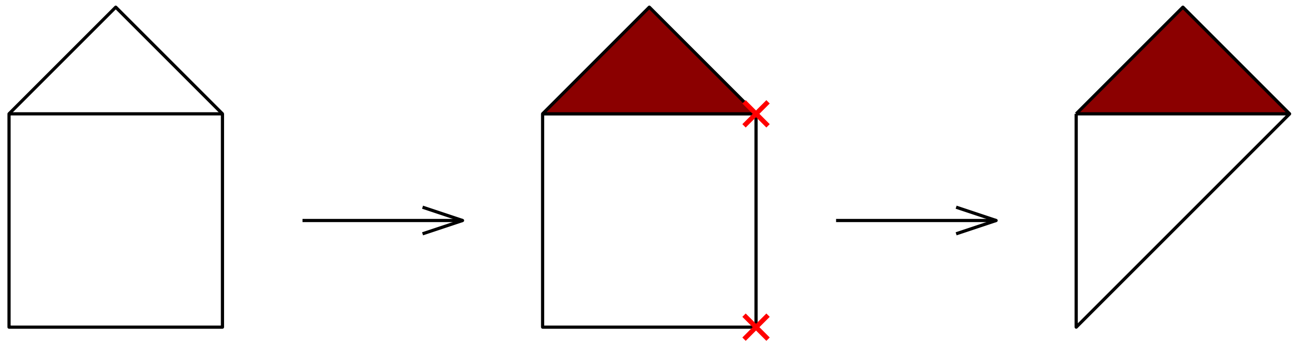
Elementary Simplicial Maps:

- inclusion of a simplex,
- contraction of two vertices.

Towers

Filtration:

$$\mathbb{K}_0 \hookrightarrow \mathbb{K}_1 \hookrightarrow \ldots \hookrightarrow \mathbb{K}_n$$



Elementary Simplicial Maps:

- inclusion of a simplex,
- contraction of two vertices.

Tower:

$$\mathbb{K}_0 \to \mathbb{K}_1 \to \ldots \to \mathbb{K}_n$$

Goal:

Transform a tower $\mathcal{T}$ into an equivalent filtration $\mathcal{F}$?

$$\mathcal{T}: \quad \mathbb{K}_0 = \emptyset \xrightarrow{\phi_0} \mathbb{K}_1 \xrightarrow{\phi_1} \dots \xrightarrow{\phi_{m-1}} \mathbb{K}_m$$

ϕ_i elementary

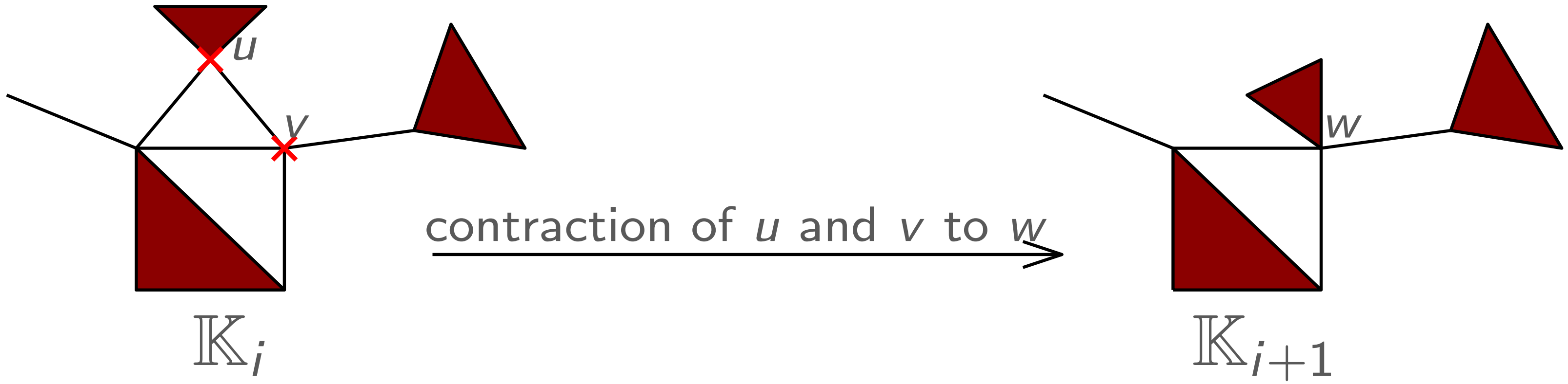
$$\Big\Downarrow \; \textbf{?}$$

$$\mathcal{F}: \quad \hat{\mathbb{K}}_0 \hookrightarrow \hat{\mathbb{K}}_1 \hookrightarrow \dots \hookrightarrow \hat{\mathbb{K}}_m$$

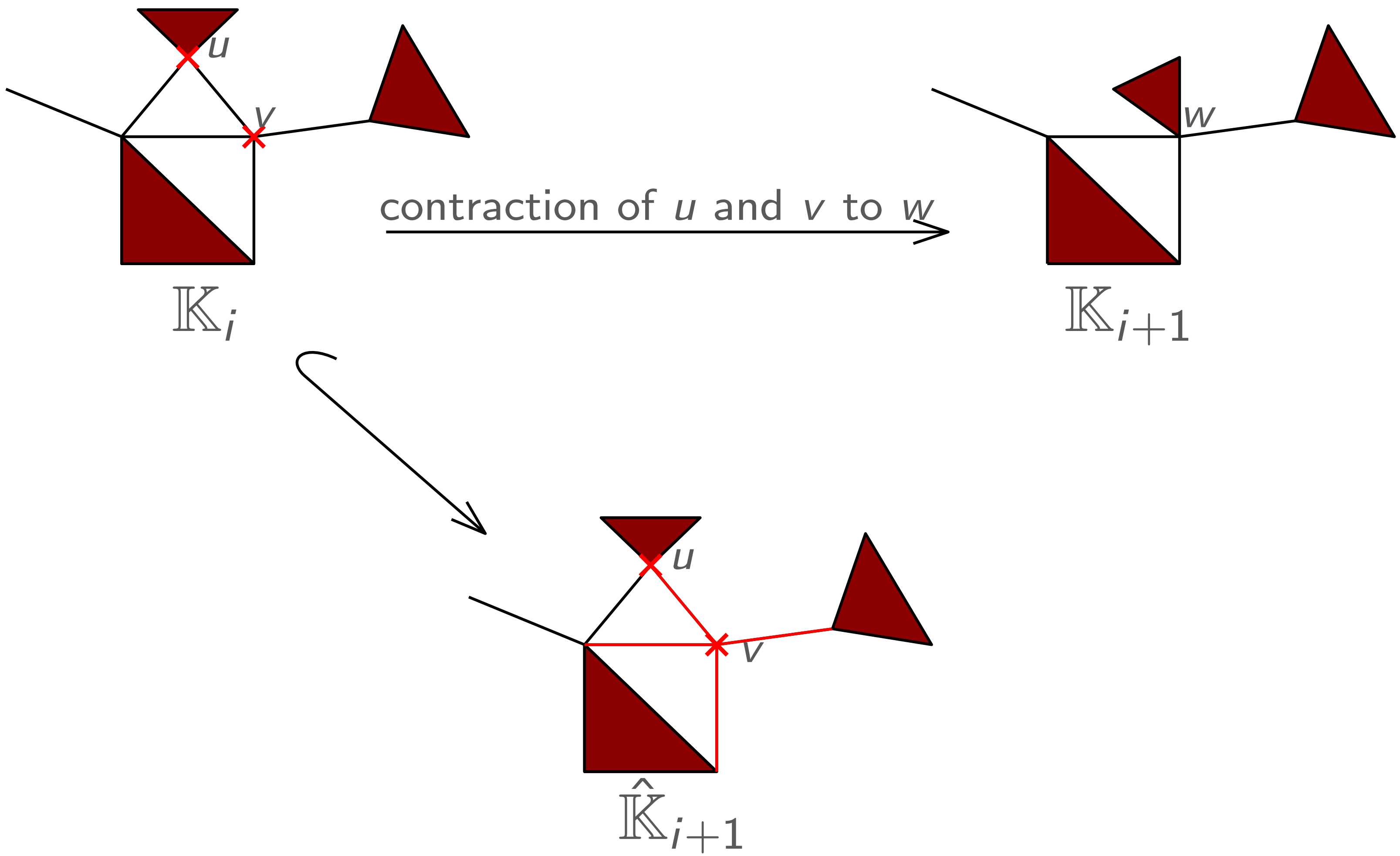
same barcode as $\mathcal{T}$

inclusions are not necessarily elementary

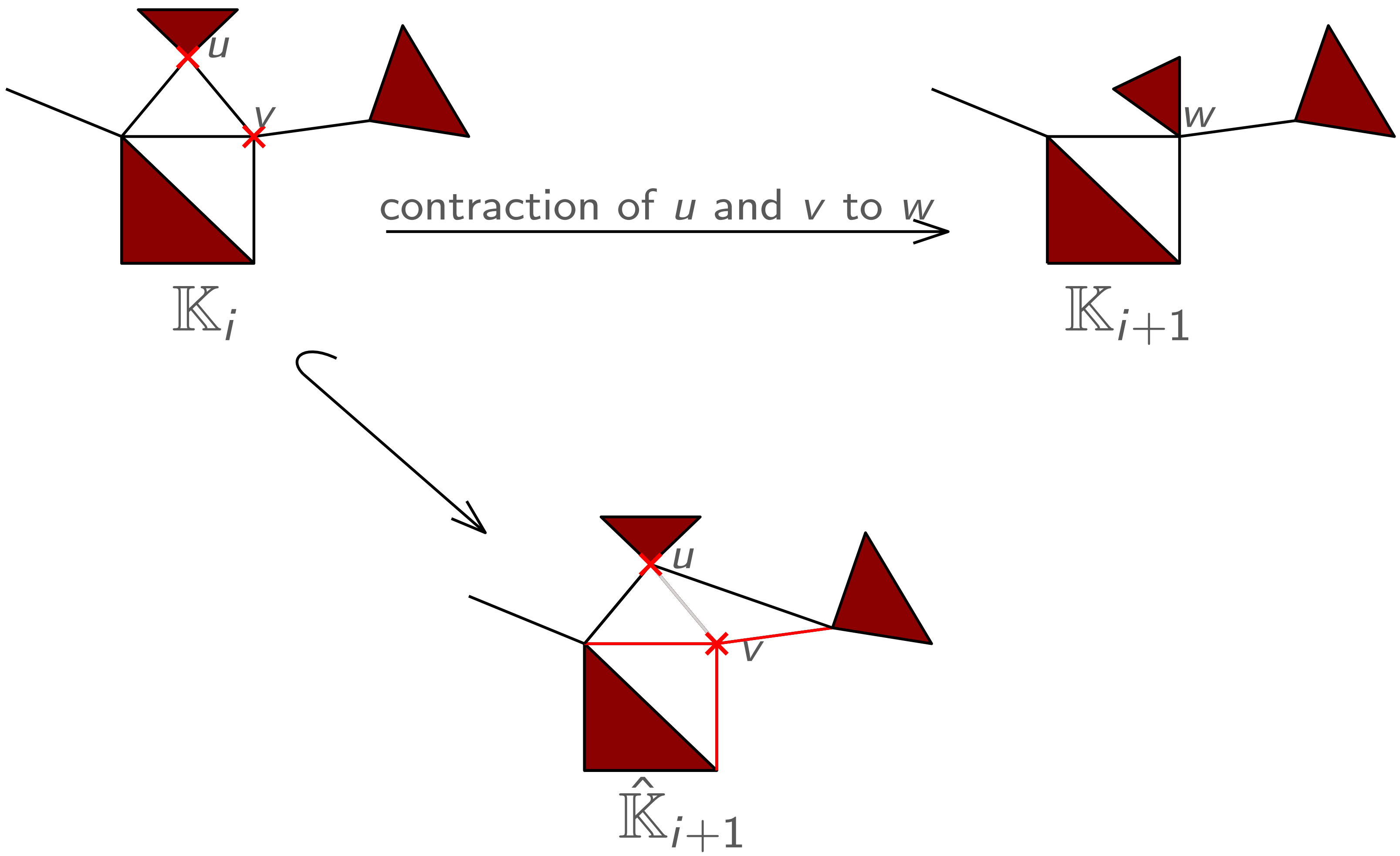
Coning Strategy [Dey, Fang & Wang]:



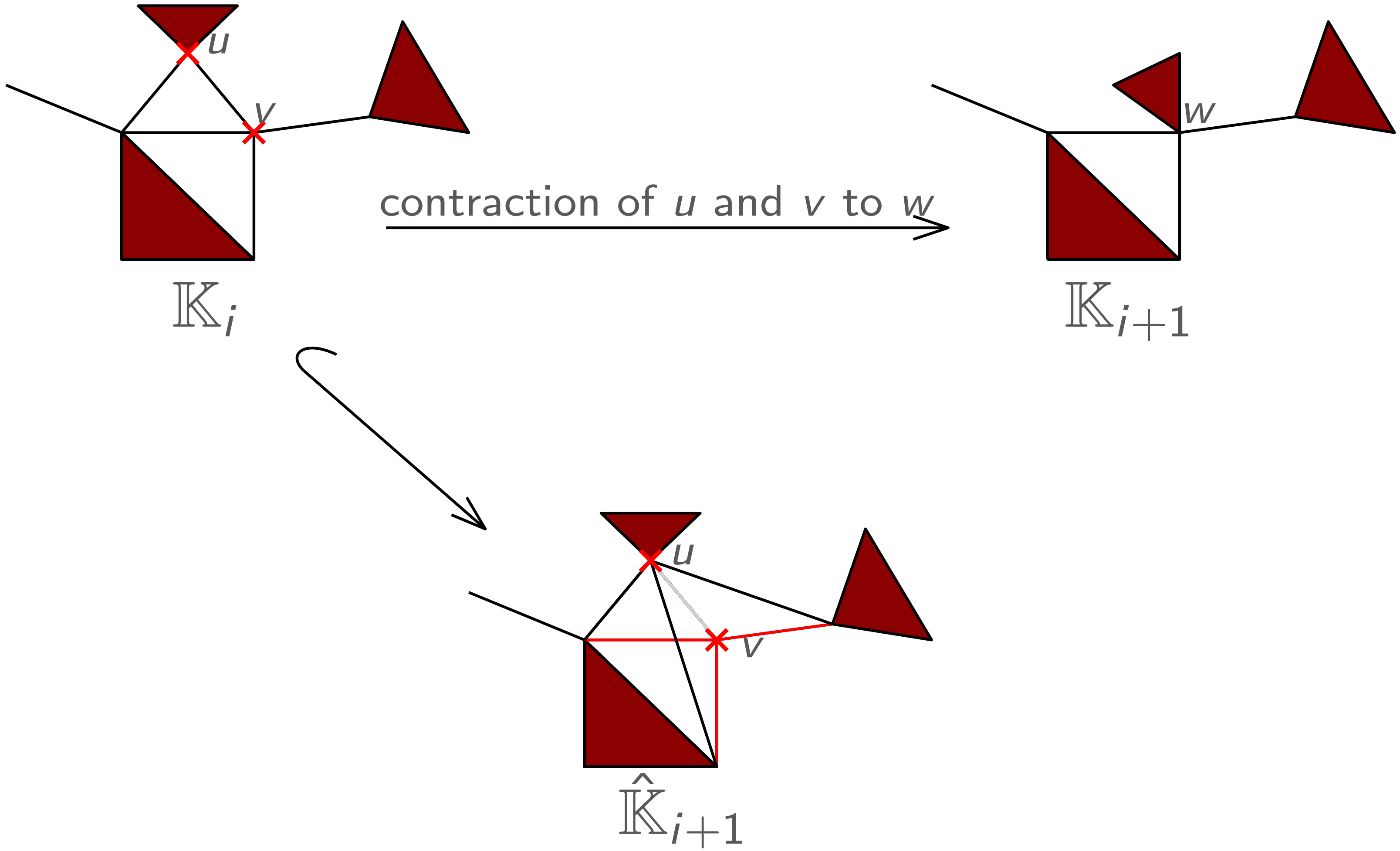
Coning Strategy [Dey, Fang & Wang]:



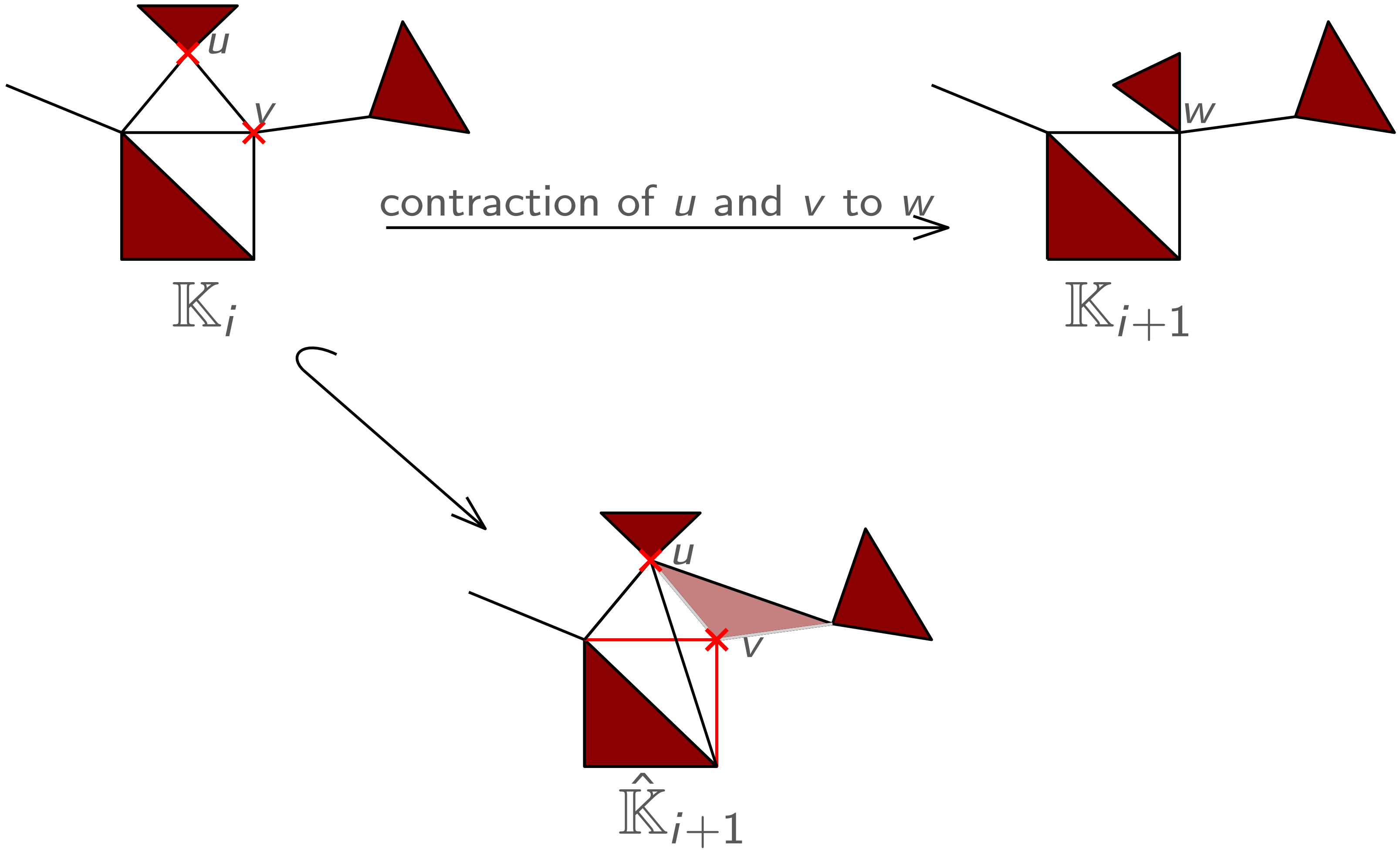
Coning Strategy [Dey, Fang & Wang]:



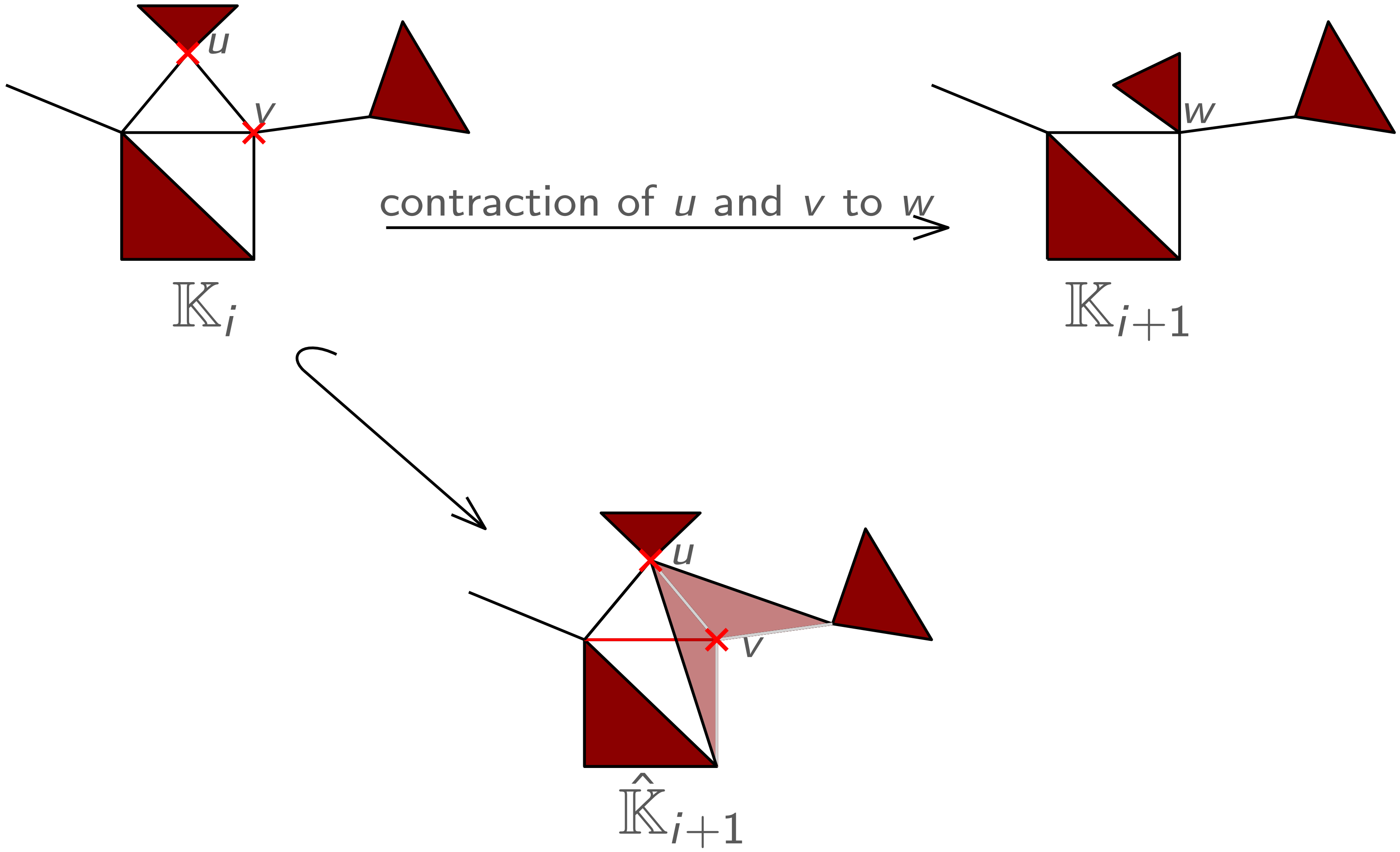
Coning Strategy [Dey, Fang & Wang]:



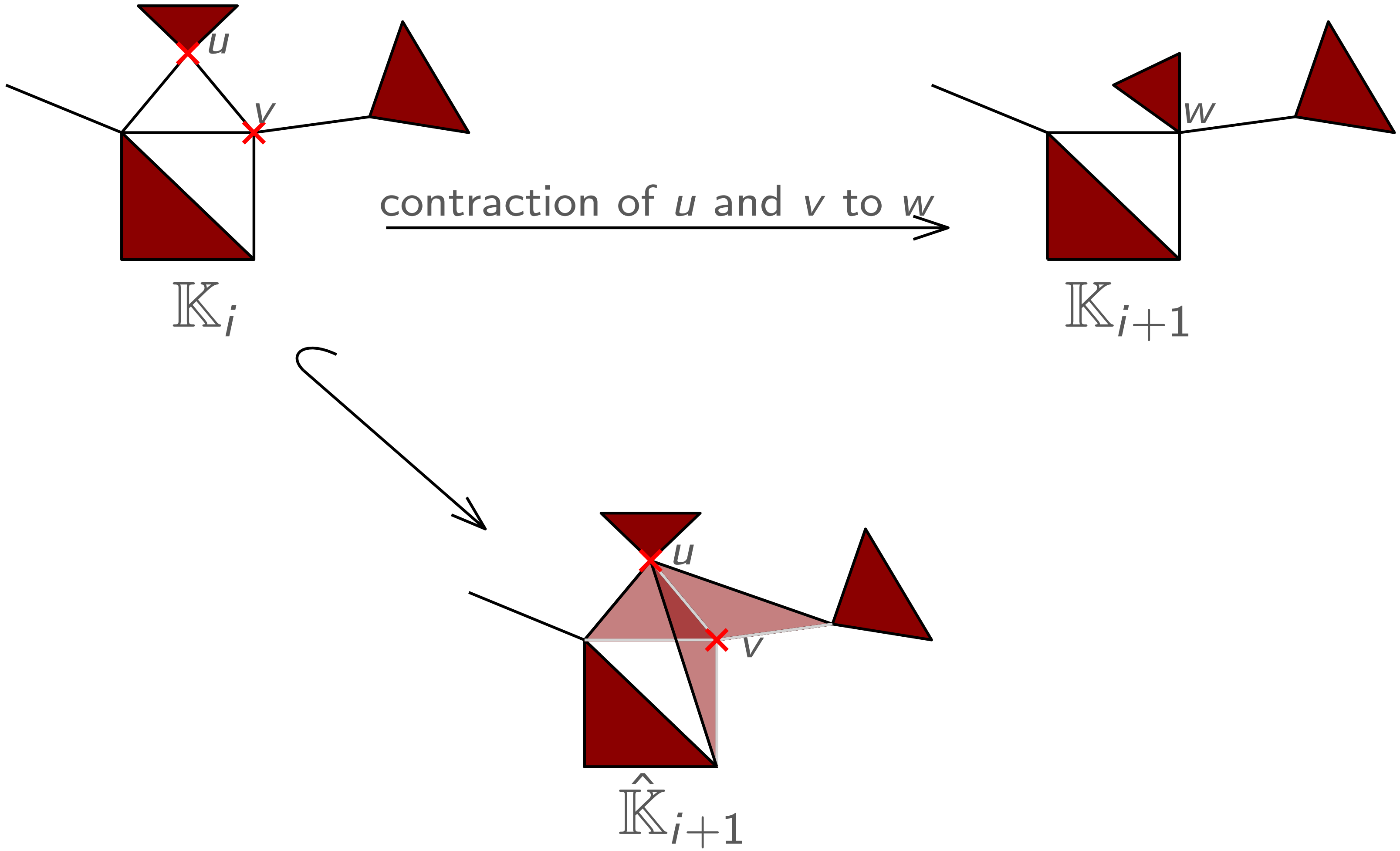
Coning Strategy [Dey, Fang & Wang]:



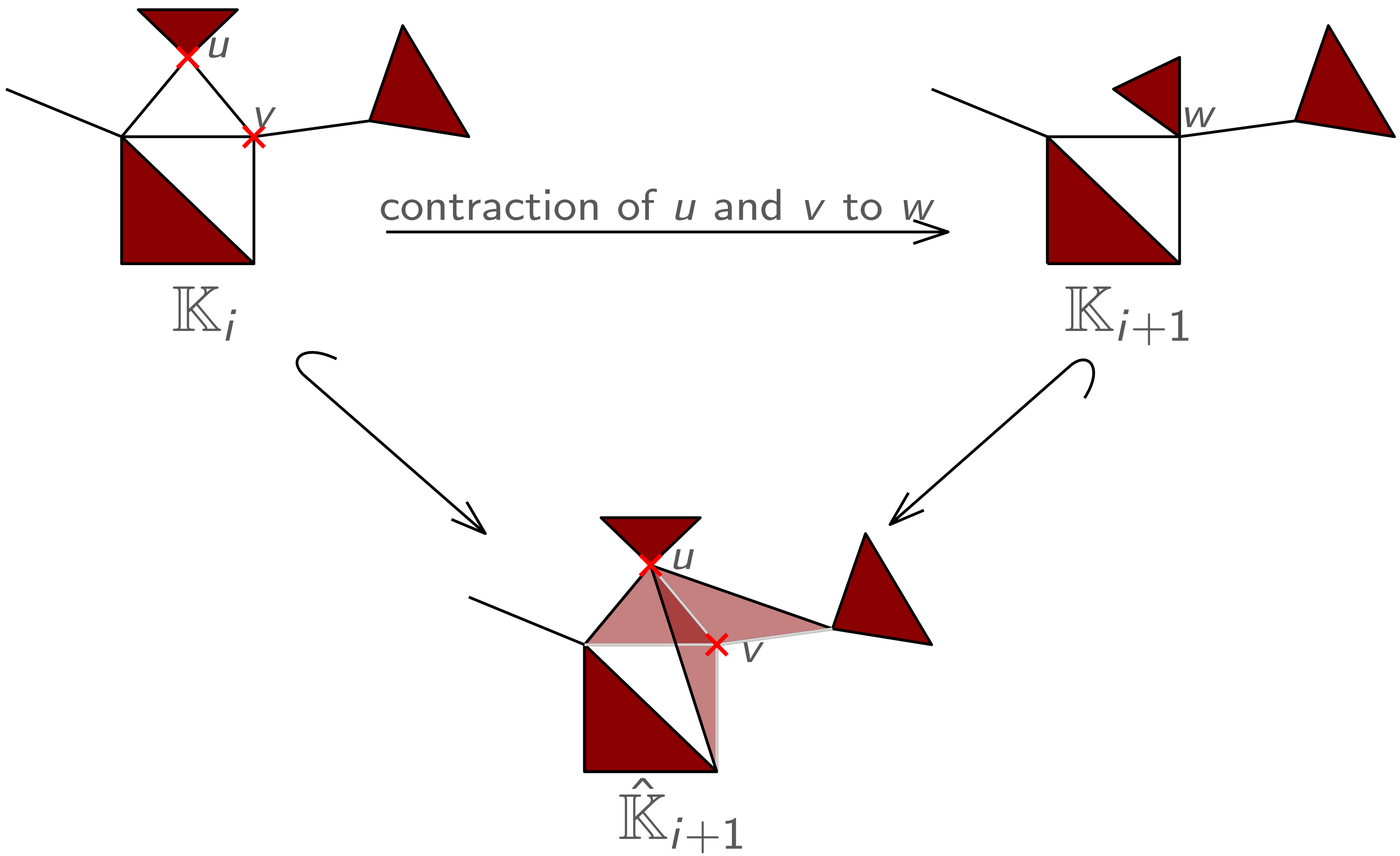
Coning Strategy [Dey, Fang & Wang]:



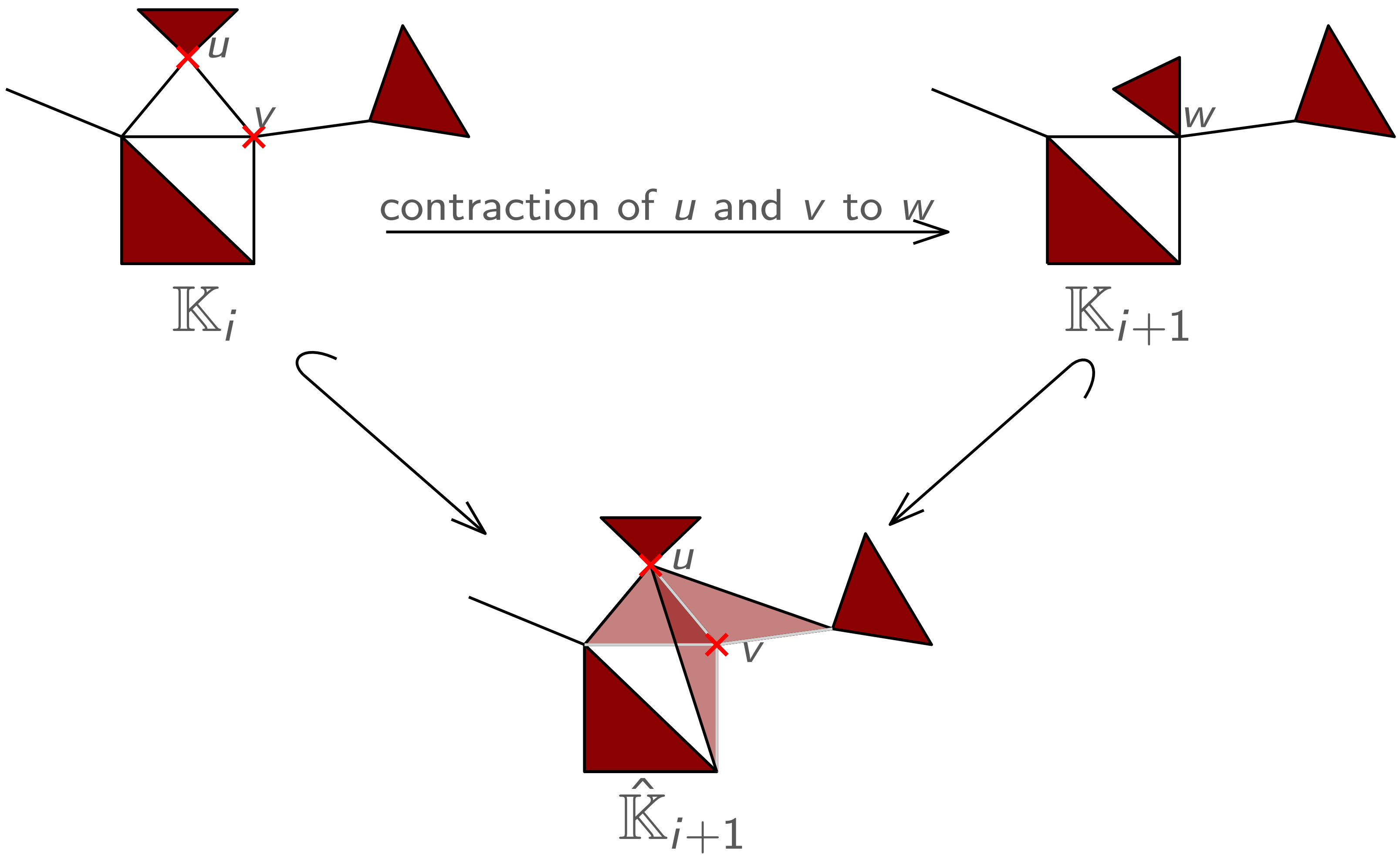
Coning Strategy [Dey, Fang & Wang]:



Coning Strategy [Dey, Fang & Wang]:



Coning Strategy [Dey, Fang & Wang]:



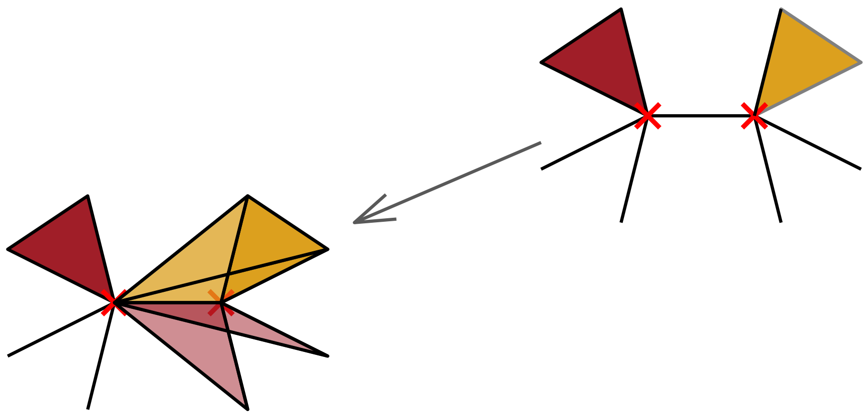
$\rightarrow$ Too expensive?

Improvements:

Construction Method

Improvements:

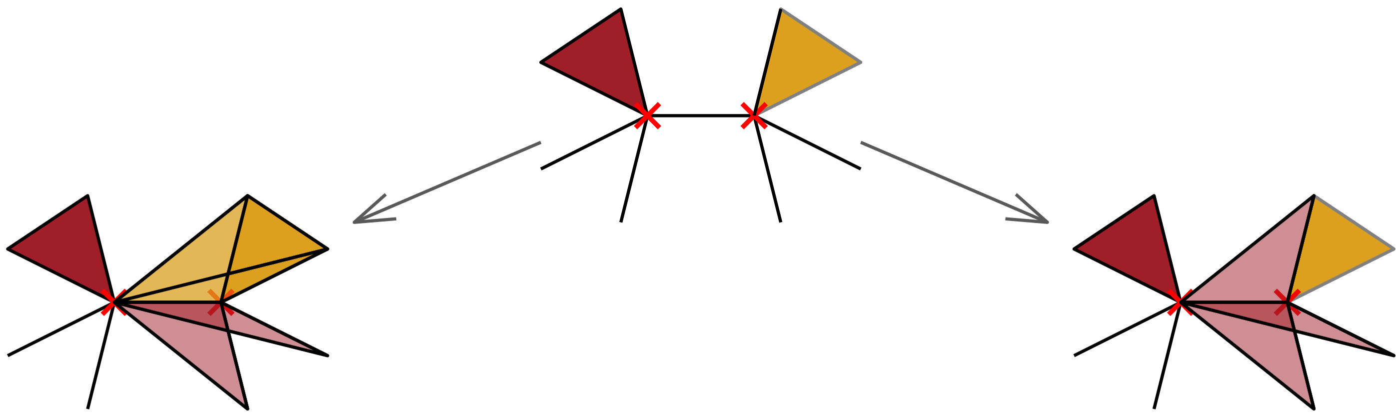
- contracted vertex and its cofaces are marked inactive,



Construction Method

Improvements:

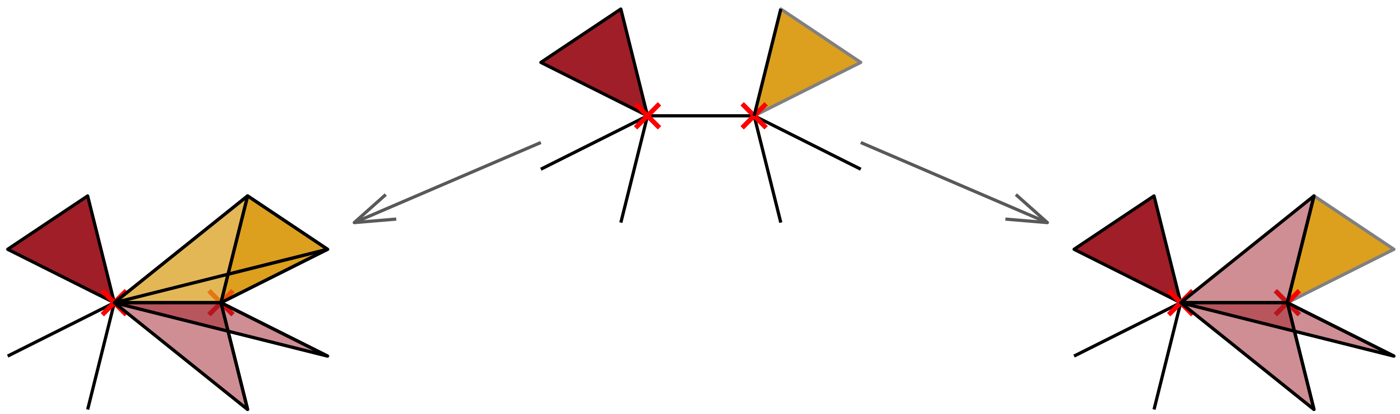
- contracted vertex and its cofaces are marked inactive,



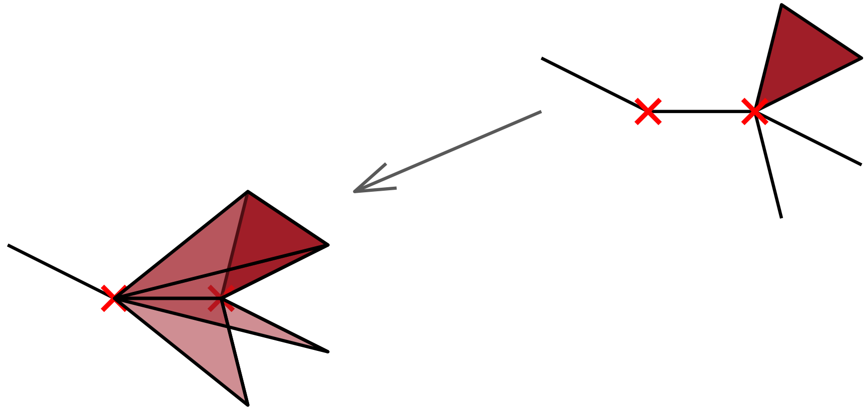
Construction Method

Improvements:

- contracted vertex and its cofaces are marked inactive,



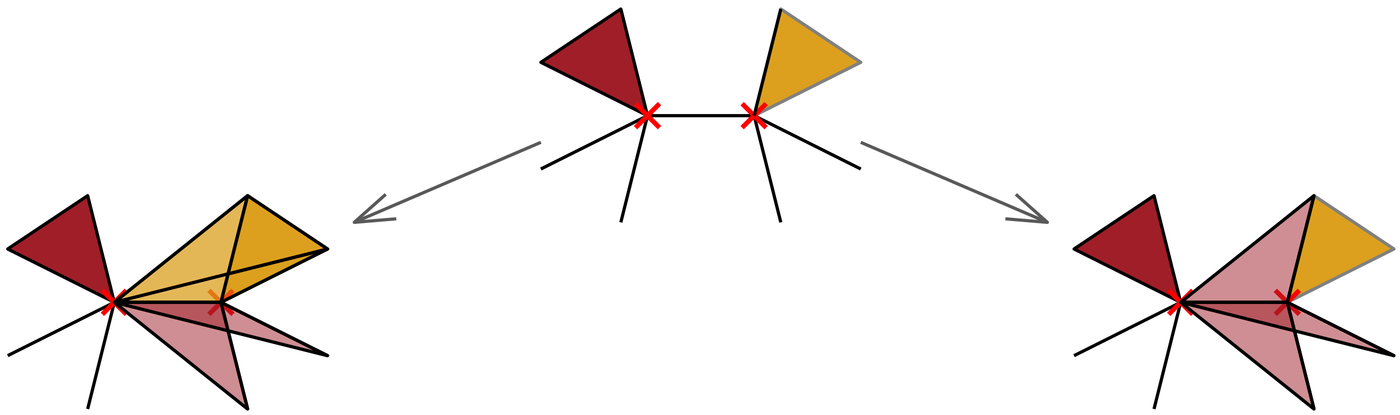
- selection of the smallest closed star.



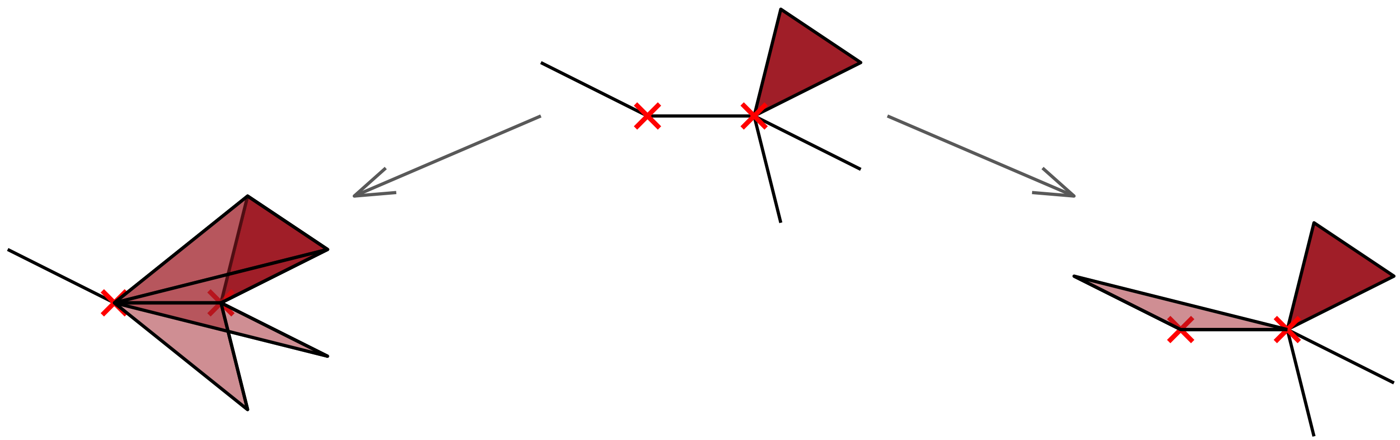
Construction Method

Improvements:

- contracted vertex and its cofaces are marked inactive,



- selection of the smallest closed star.



Result

$$\mathcal{T}: \quad \mathbb{K}_0 = \emptyset \xrightarrow{\phi_0} \mathbb{K}_1 \xrightarrow{\phi_1} \dots \xrightarrow{\phi_{m-1}} \mathbb{K}_m$$

ϕ_i elementary

$$\Big\downarrow$$

$$\mathcal{F}: \quad \hat{\mathbb{K}}_0 \hookrightarrow \hat{\mathbb{K}}_1 \hookrightarrow \dots \hookrightarrow \hat{\mathbb{K}}_m$$

same barcode as $\mathcal{T}$

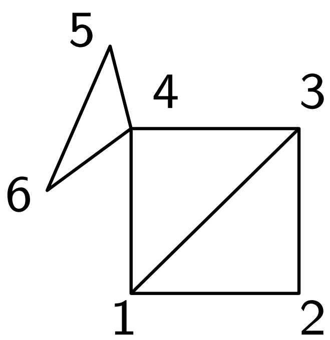
inclusions are not necessarily elementary

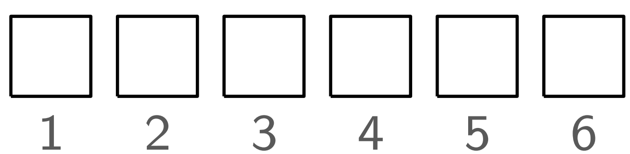
$n \leq m$: total number of elementary inclusions,
Δ: dimension of $\mathcal{T}$:

Theorem. $|\hat{\mathbb{K}}_m|$ is at most $O(\Delta \cdot n \log n)$.

Idea of Proof

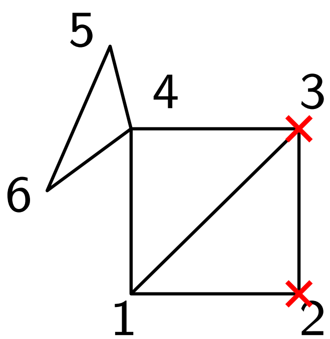
Contracting Forest:

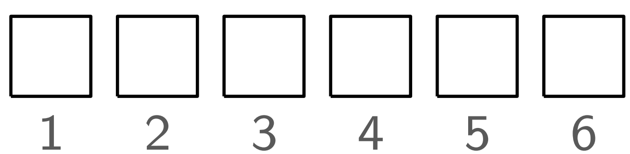




Idea of Proof

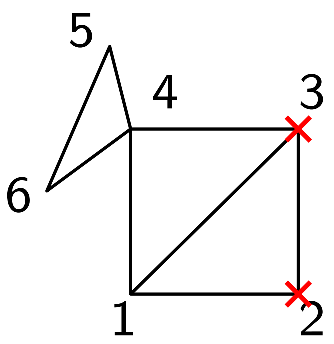
Contracting Forest:

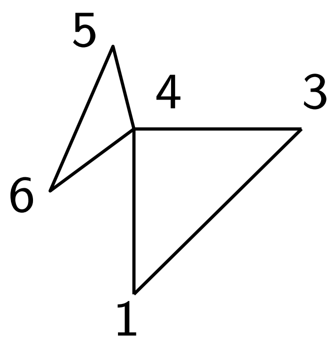


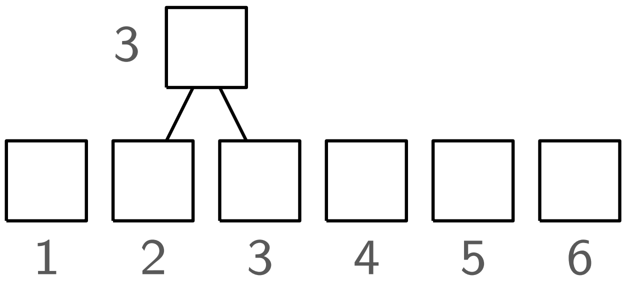


Idea of Proof

Contracting Forest:

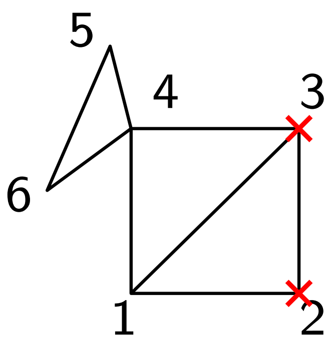


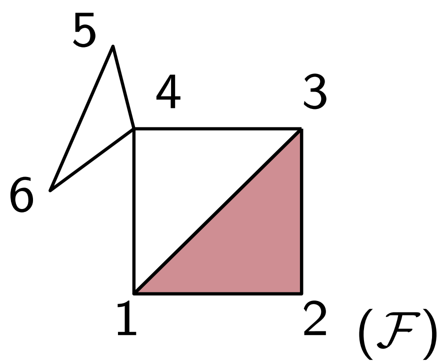


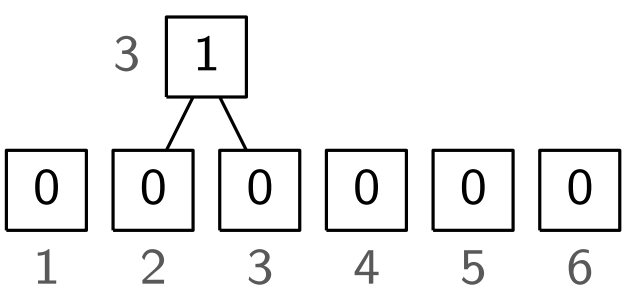


Idea of Proof

Contracting Forest:

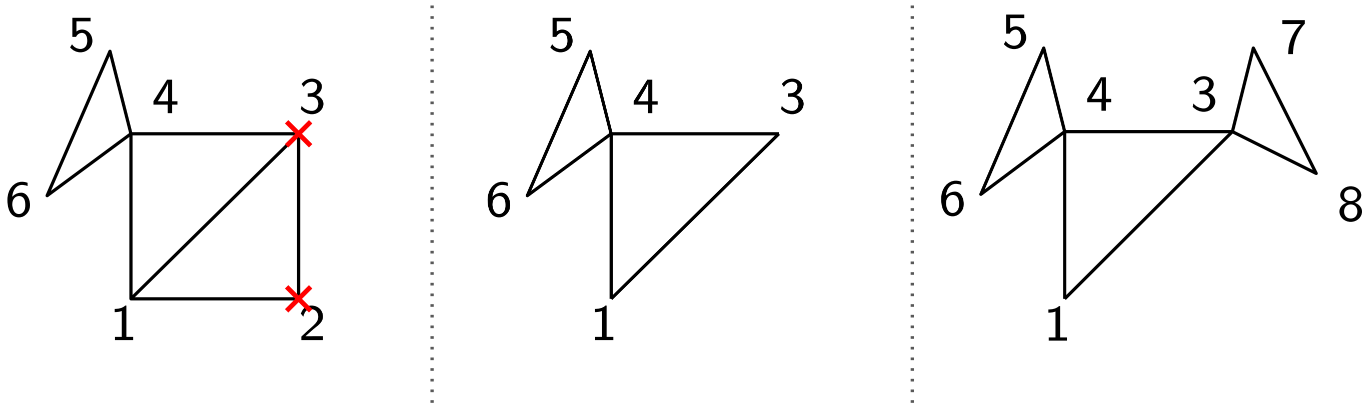


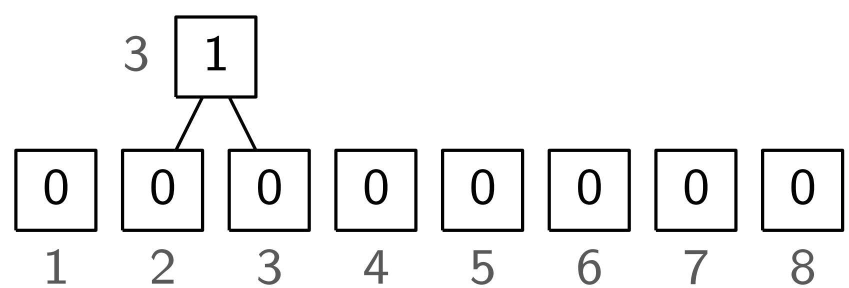




Idea of Proof

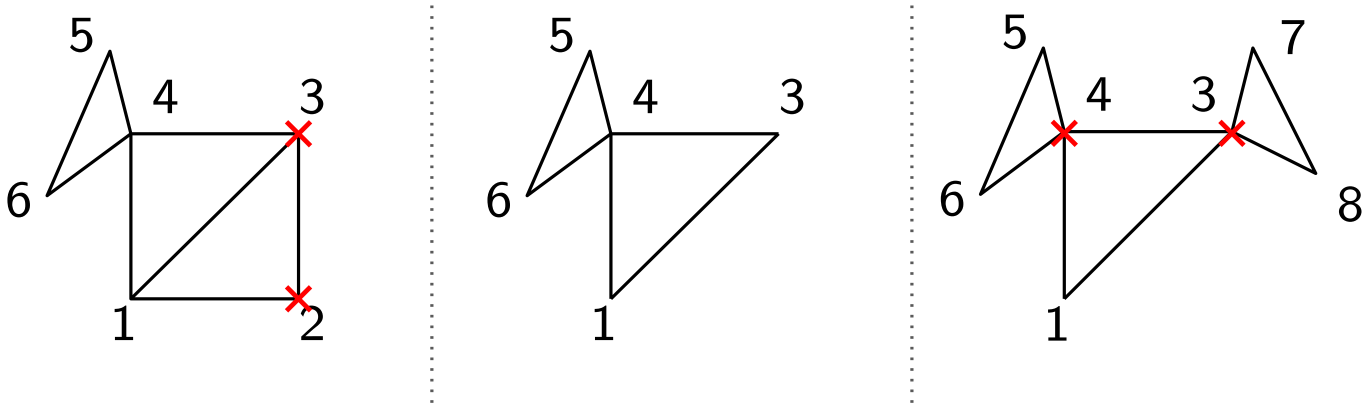
Contracting Forest:

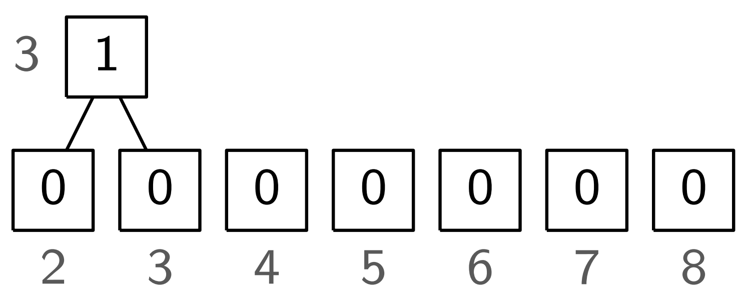




Idea of Proof

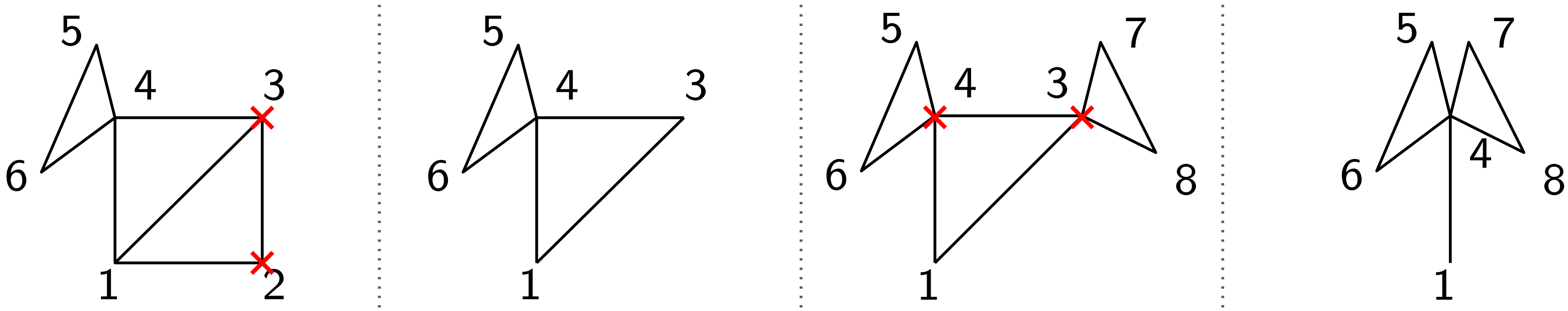
Contracting Forest:

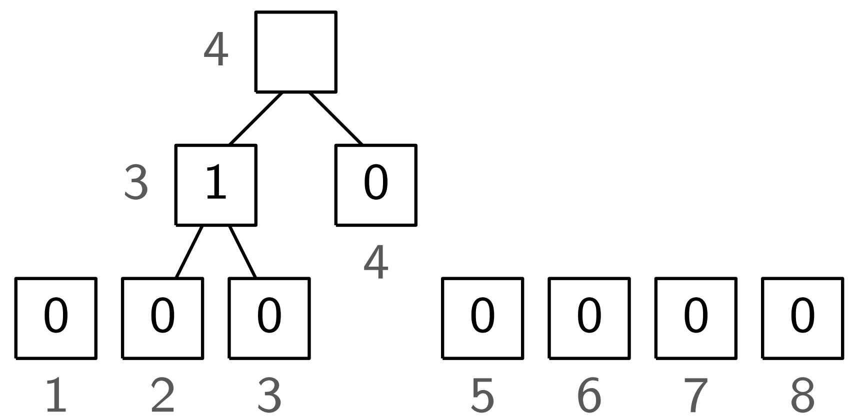




Idea of Proof

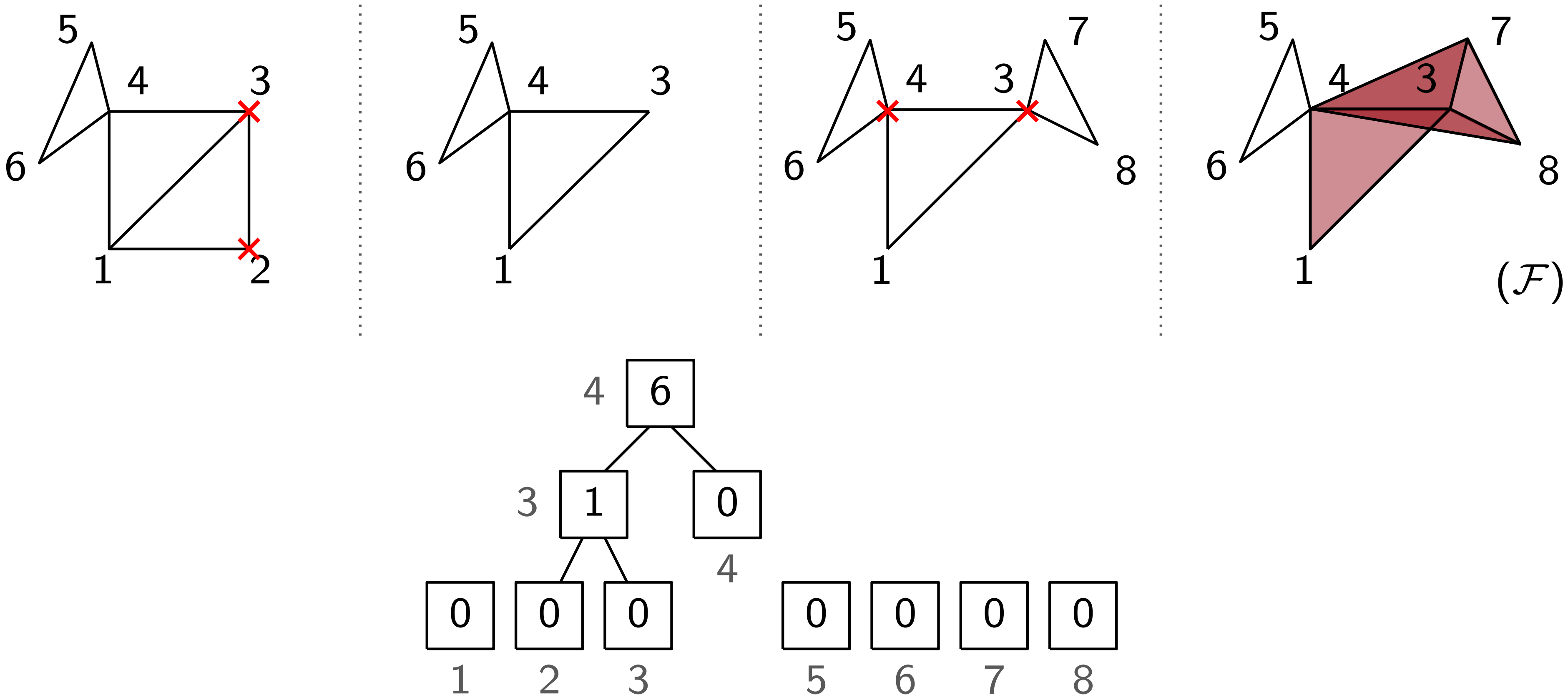
Contracting Forest:





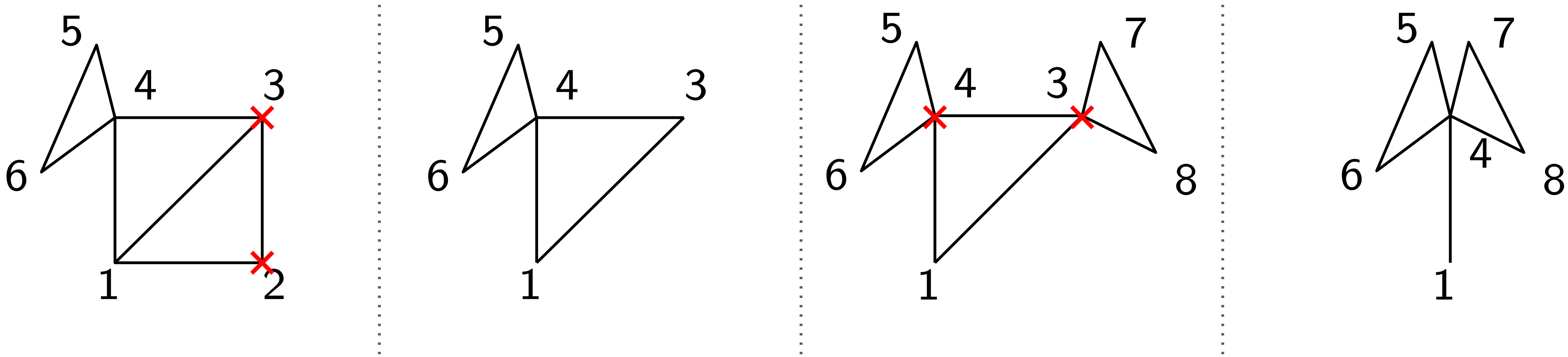
Idea of Proof

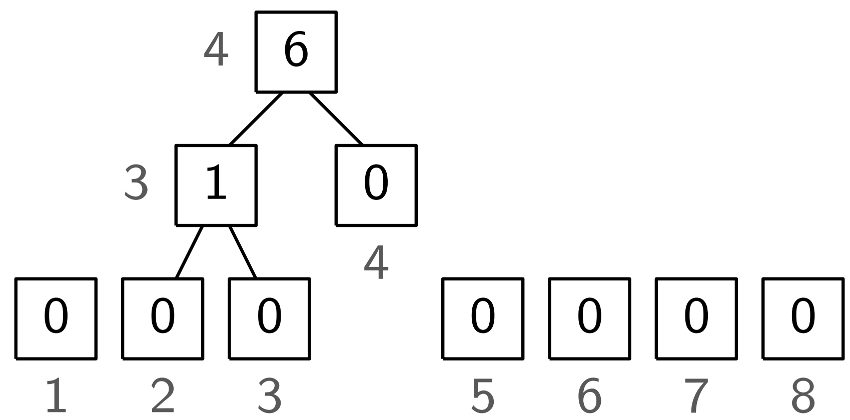
Contracting Forest:



Idea of Proof

Contracting Forest:



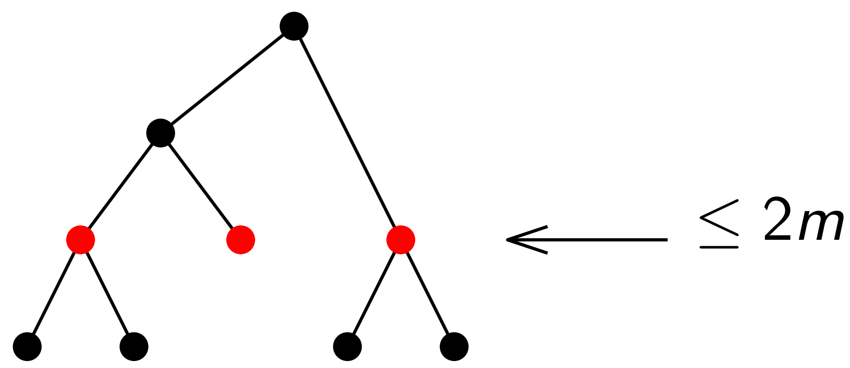


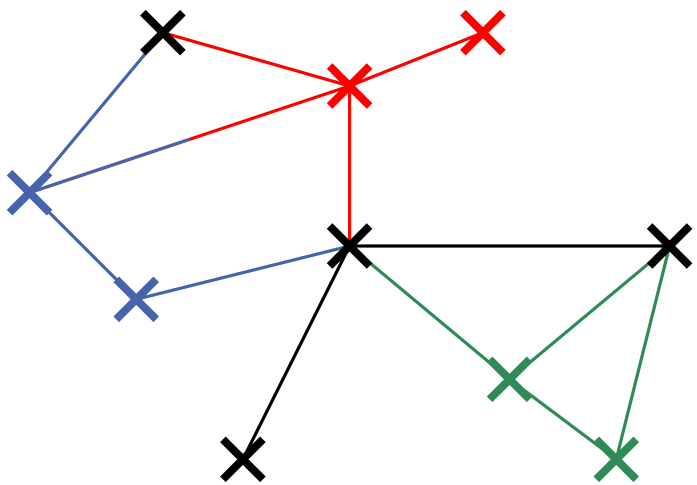
$\rightarrow$ Sum of all nodes $=$ total additional length of $\mathcal{F}$

Idea of Proof

m: number of edges inserted in $\mathcal{T}$, $\dim \mathcal{T} = 1$

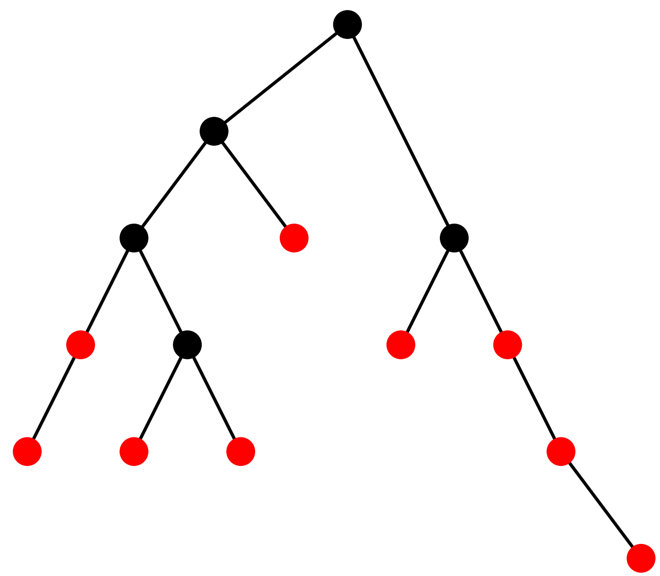
cost = minimum number of adjacent edges





Idea of Proof

n: number of insertions in $\mathcal{T}$



$$\Sigma \leq 2 \cdot (\Delta + 1) \cdot n$$

$\rightarrow$ maximal $\log n$ such sets,
$\rightarrow$ total of $O(\Delta \cdot n \log n)$.

Experimental Results

		Alg1 + PHAT			Simpers (Dey et al.)	
	tower size	filtration size	time (s)	mem. peak (kB)	time (s)	mem. peak (kB)
data1	5 328	19 747	0.12	7 040	2.49	10 188
data2	9 773	35 253	0.20	10 228	13.97	20 308
data3	9 237	38 101	0.22	10 916	19.29	24 924
GPS	10 331	9 063	0.07	5 292	0.35	6 064
KB	117 518	133 433	0.50	18 712	2.83	24 460
MC	167 002	185 447	0.72	25 272	4.12	26 020
S3	1 726 575	1 824 461	10.09	221 636	49.86	239 404
PC25	10 261 124	12 283 003	135.02	2 223 544	∞	-

ω: width of the tower = max. size of a complex in $\mathcal{T}$,
C_ω: cost of an operation in a dictionary with ω elements:

Theorem. $\mathcal{F}$ can be computed from $\mathcal{T}$ with a streaming algorithm in $O(\Delta \cdot |\hat{\mathbb{K}}_m| \cdot C_\omega)$ time and space complexity $O(\Delta \cdot \omega)$.

Experimental Results

		Alg1 + PHAT			Simpers (Dey et al.)	
	tower size	filtration size	time (s)	mem. peak (kB)	time (s)	mem. peak (kB)
data1	5 328	19 747	0.12	7 040	2.49	10 188
data2	9 773	35 253	0.20	10 228	13.97	20 308
data3	9 237	38 101	0.22	10 916	19.29	24 924
GPS	10 331	9 063	0.07	5 292	0.35	6 064
KB	117 518	133 433	0.50	18 712	2.83	24 460
MC	167 002	185 447	0.72	25 272	4.12	26 020
S3	1 726 575	1 824 461	10.09	221 636	49.86	239 404
PC25	10 261 124	12 283 003	135.02	2 223 544	∞	-

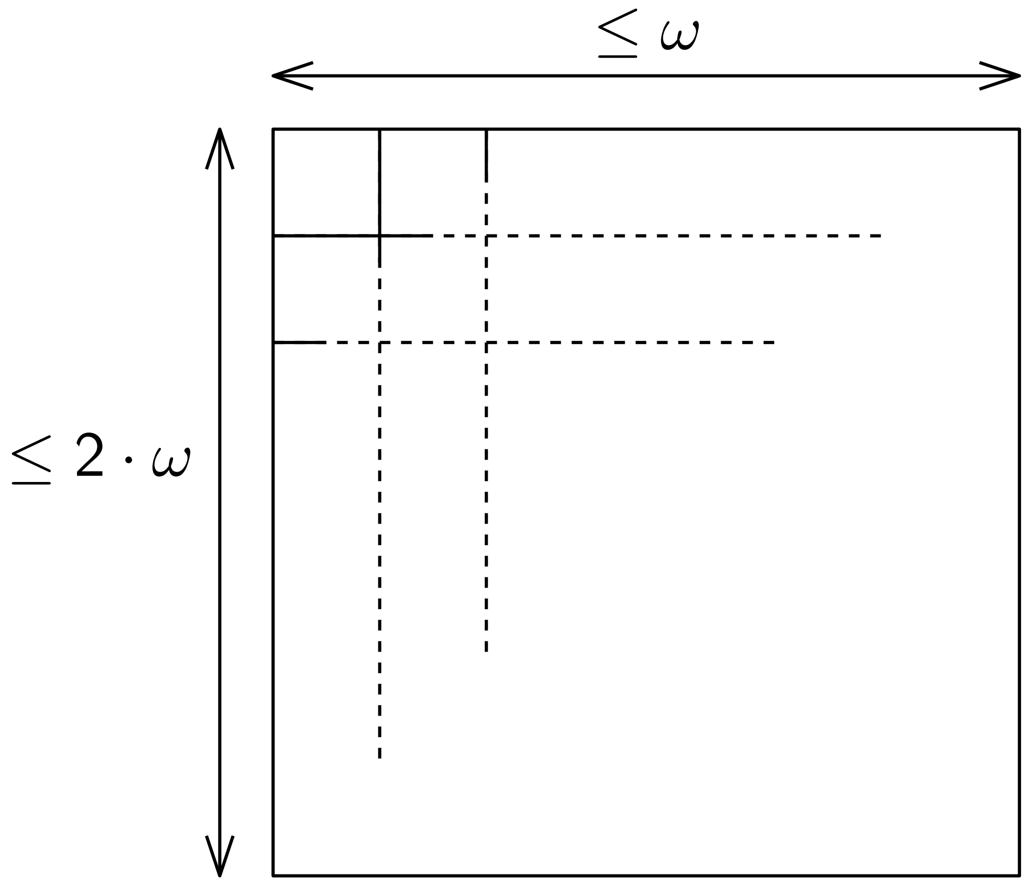
ω: width of the tower = max. size of a complex in $\mathcal{T}$,
C_ω: cost of an operation in a dictionary with ω elements:

Theorem. $\mathcal{F}$ can be computed from $\mathcal{T}$ with a streaming algorithm in $O(\Delta \cdot |\hat{\mathbb{K}}_m| \cdot C_\omega)$ time and space complexity $O(\Delta \cdot \omega)$.

$\rightarrow$ Computing barcodes still uses too much memory.

Streaming Algorithm for Persistent Homology

ω: max. size of a complex in $\mathcal{T}$



boundary matrix

References:

- U. Bauer, M. Kerber, and J. Reininghaus. Clear and compress: Computing persistent homology in chunks. In *Topological Methods in Data Analysis and Visualization III*, Mathematics and Visualization, pages 103–117. Springer, 2014.
- C. Chen and M. Kerber. Persistent homology computation with a twist. In *European Workshop on Computational Geometry (EuroCG)*, pages 197–200, 2011.
- T. Dey, F. Fan, and Y. Wang. Computing Topological Persistence for Simplicial Maps. In *ACM Symposium on Computational Geometry (SoCG)*, pages 345–354, 2014.
- A. Zomorodian and G. Carlsson. Computing persistent homology. *Discrete & Computational Geometry*, 33(2):249–274, 2005.

Conclusion

References:

- U. Bauer, M. Kerber, and J. Reininghaus. Clear and compress: Computing persistent homology in chunks. In *Topological Methods in Data Analysis and Visualization III*, Mathematics and Visualization, pages 103–117. Springer, 2014.
- C. Chen and M. Kerber. Persistent homology computation with a twist. In *European Workshop on Computational Geometry (EuroCG)*, pages 197–200, 2011.
- T. Dey, F. Fan, and Y. Wang. Computing Topological Persistence for Simplicial Maps. In *ACM Symposium on Computational Geometry (SoCG)*, pages 345–354, 2014.
- A. Zomorodian and G. Carlsson. Computing persistent homology. *Discrete & Computational Geometry*, 33(2):249–274, 2005.

Thank you for your attention!