A Schnyder-type drawing algorithm for 5-connected triangulations

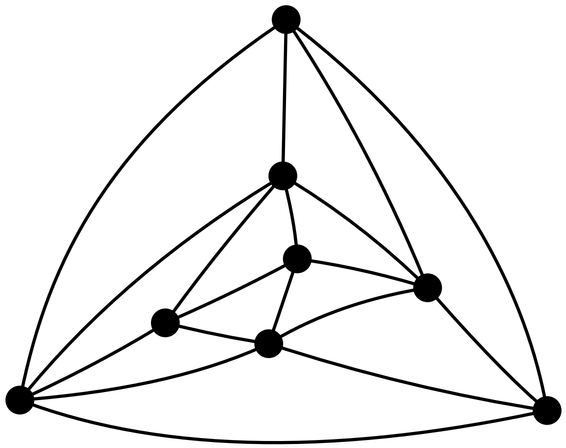
Olivier Bernardi[1], Éric Fusy[2] and Shizhe Liang[1]

1. Dept. of Math, Brandeis University
2. LIGM/CNRS, Université Gustave Eiffel

Graph drawing'23

Triangulations

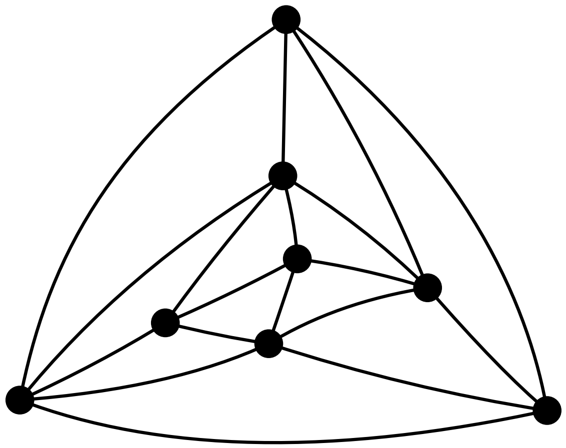
Triangulation = graph embedded in the plane, all faces of degree 3



= embedded maximal planar graph

Triangulations

Triangulation = graph embedded in the plane, all faces of degree 3



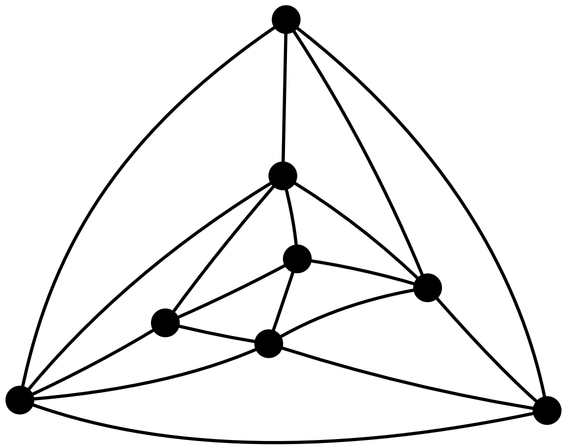
= embedded maximal planar graph

For any graph (def):

k-connected: deleting any subset of $< k$ vertices does not disconnect

Triangulations

Triangulation = graph embedded in the plane, all faces of degree 3



= embedded maximal planar graph

For any graph (def):

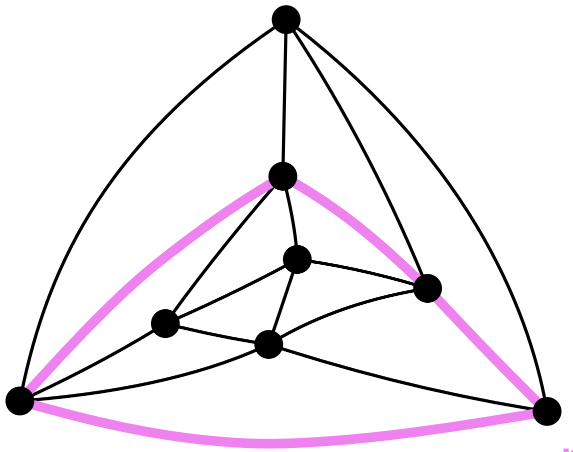
k-connected: deleting any subset of $< k$ vertices does not disconnect

For triangulations:

3-connected	$\Leftrightarrow$	simple
4-connected	$\Leftrightarrow$	no separating 3-cycle
5-connected	$\Leftrightarrow$	no separating 4-cycle

Triangulations

Triangulation = graph embedded in the plane, all faces of degree 3



= embedded maximal planar graph

4-connected
not 5-connected

For any graph (def):

k-connected: deleting any subset of $< k$ vertices does not disconnect

For triangulations:

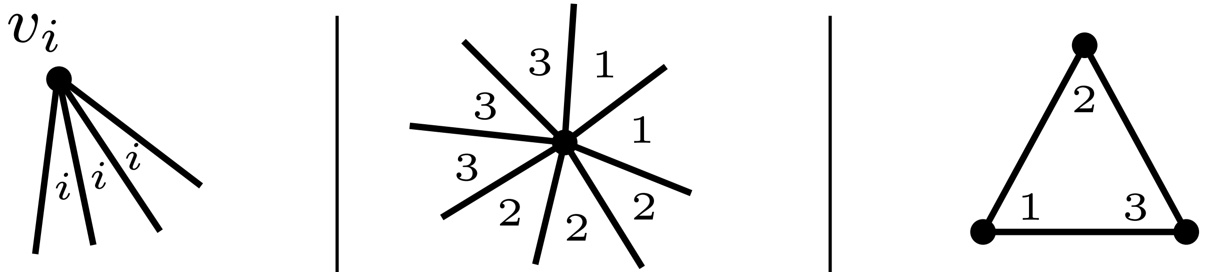
3-connected	$\Leftrightarrow$	simple
4-connected	$\Leftrightarrow$	no separating 3-cycle
5-connected	$\Leftrightarrow$	no separating 4-cycle

3-connected case

Schnyder structures on simple triangulations

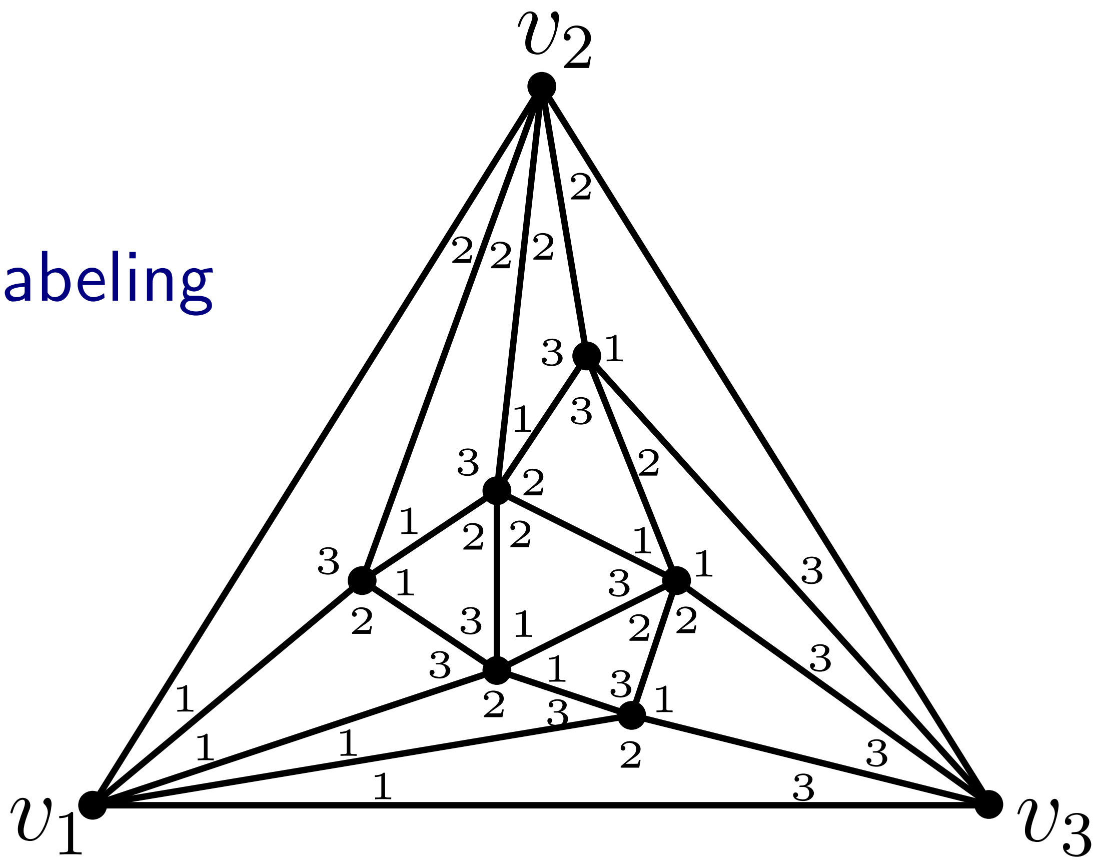
[Schnyder'89]

Any triangulation admits a **labeling** of corners by $\{1, 2, 3\}$ satisfying



outer vertices | inner vertices | inner faces

Schnyder labeling

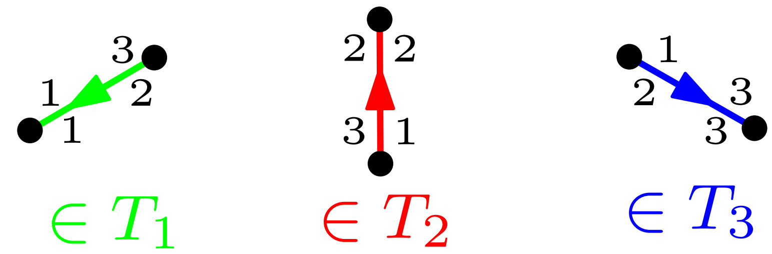


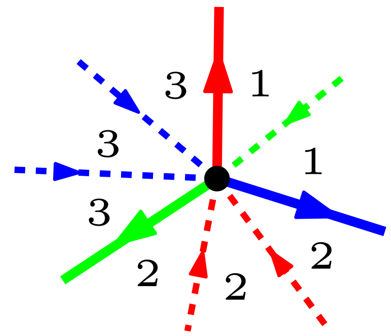
Schnyder structures on simple triangulations

[Schnyder'89]

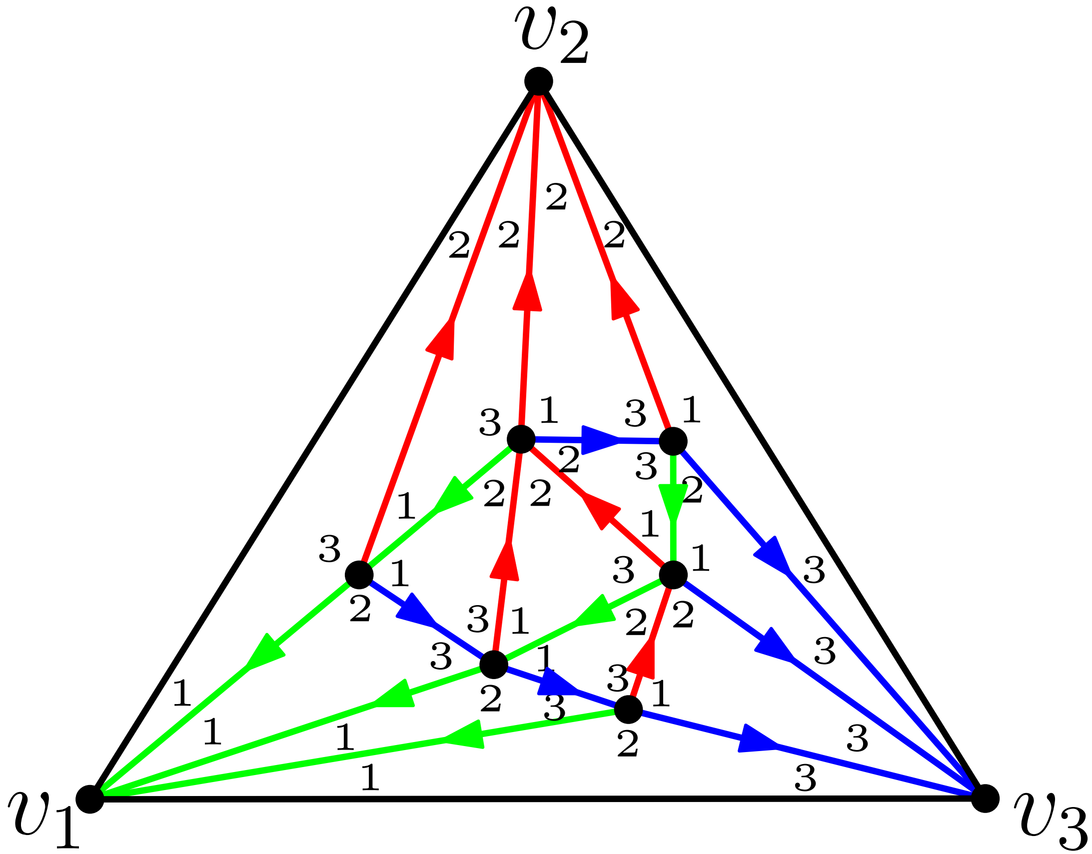
Yields 3 spanning trees T_1, T_2, T_3 (Schnyder wood)

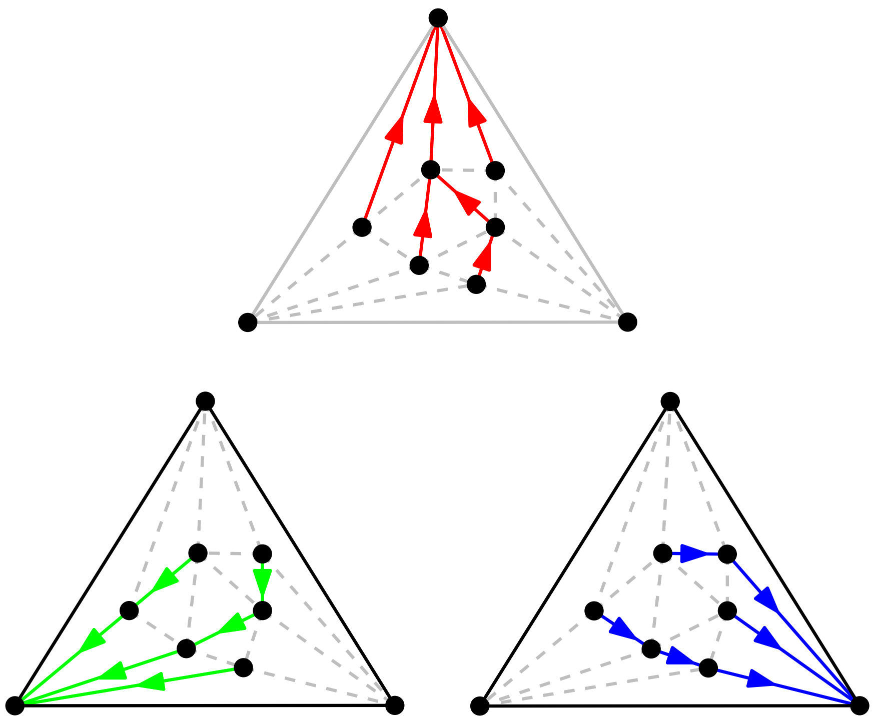
3 types of edges



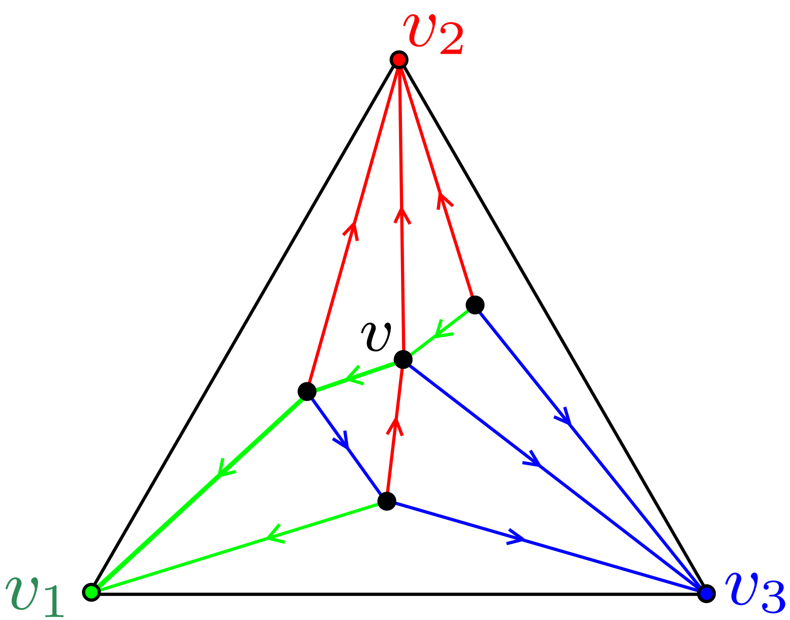


inner vertex



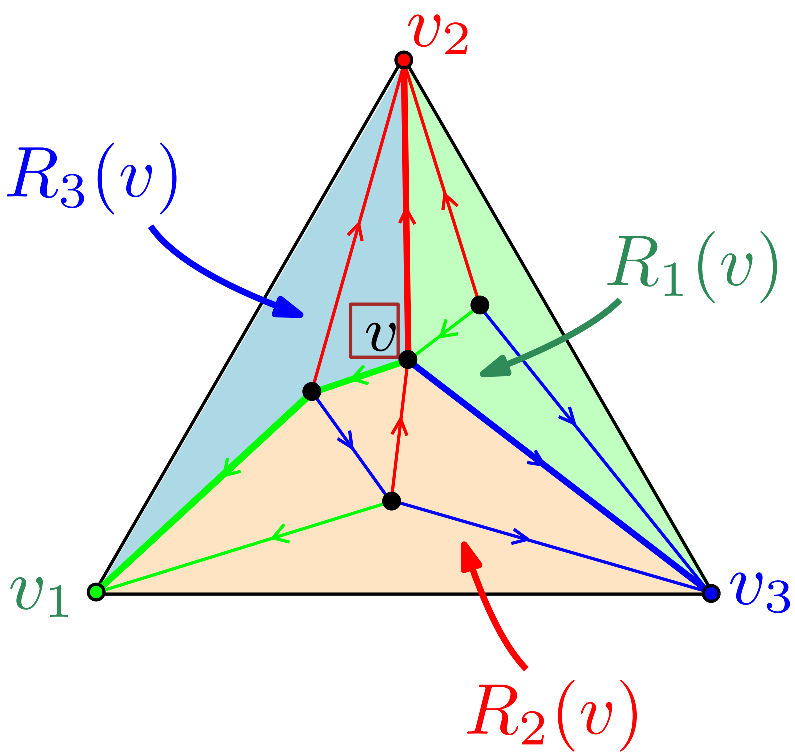


Schnyder's face-counting algorithm [Schnyder'90]



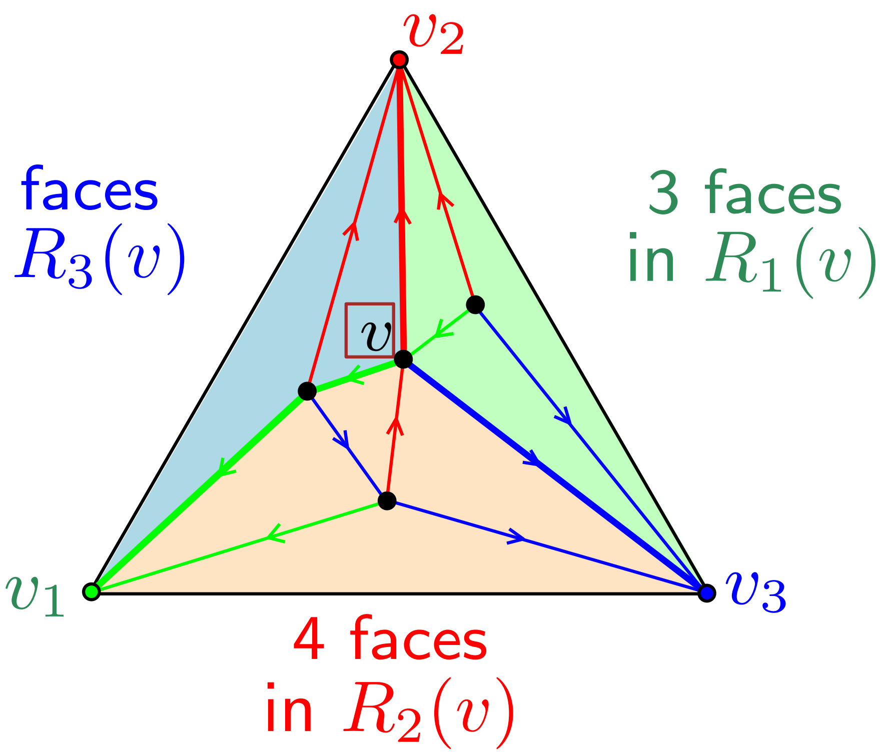
Schnyder's face-counting algorithm [Schnyder'90]

tree-paths from v partition inner faces into
3 regions $R_1(v)$, $R_2(v)$, $R_3(v)$



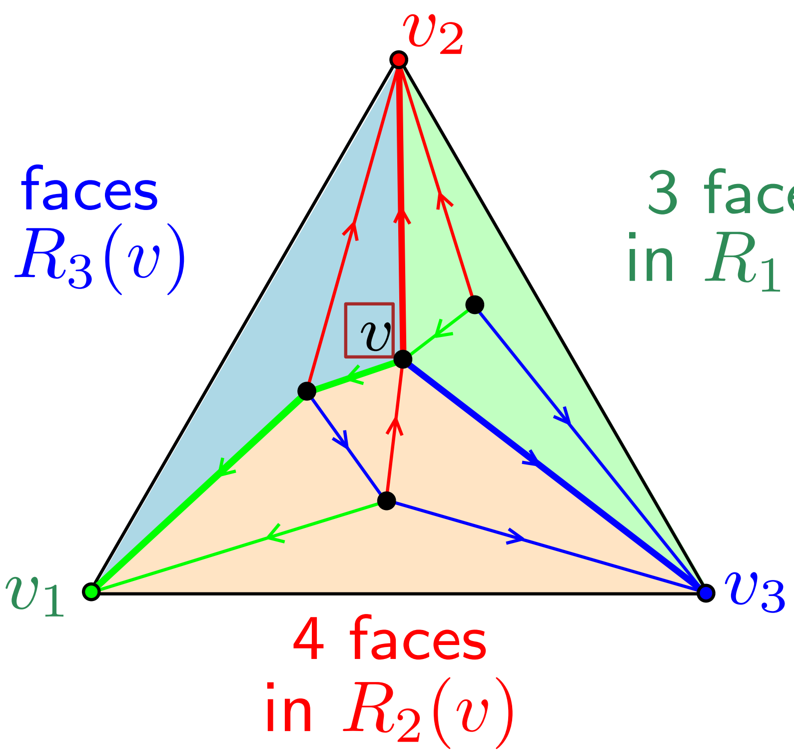
Schnyder's face-counting algorithm [Schnyder'90]

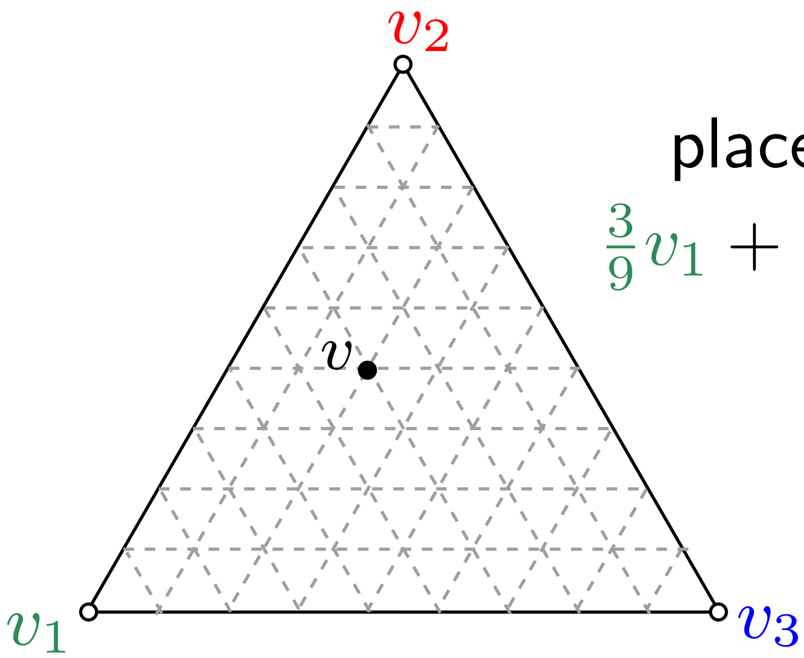
tree-paths from v partition inner faces into
3 regions $R_1(v)$, $R_2(v)$, $R_3(v)$



Schnyder's face-counting algorithm [Schnyder'90]

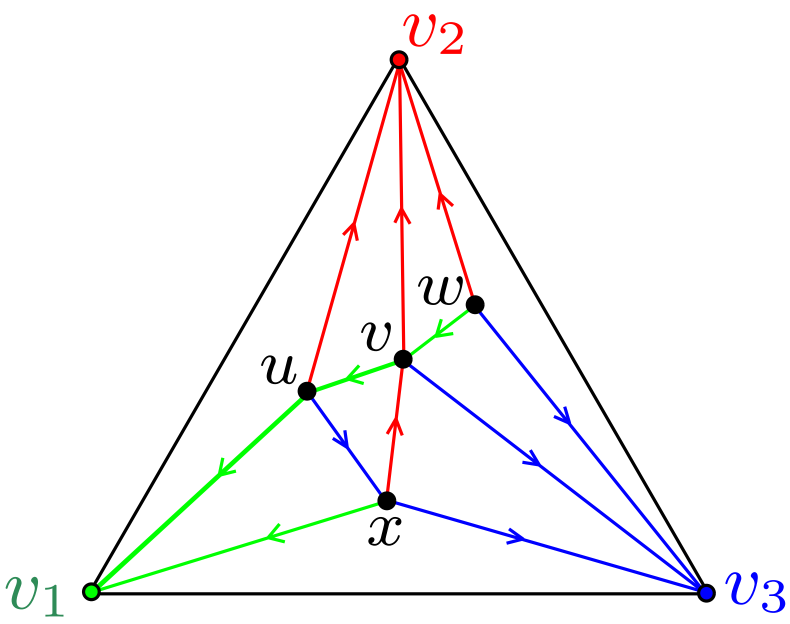
tree-paths from v partition inner faces into
3 regions $R_1(v)$, $R_2(v)$, $R_3(v)$

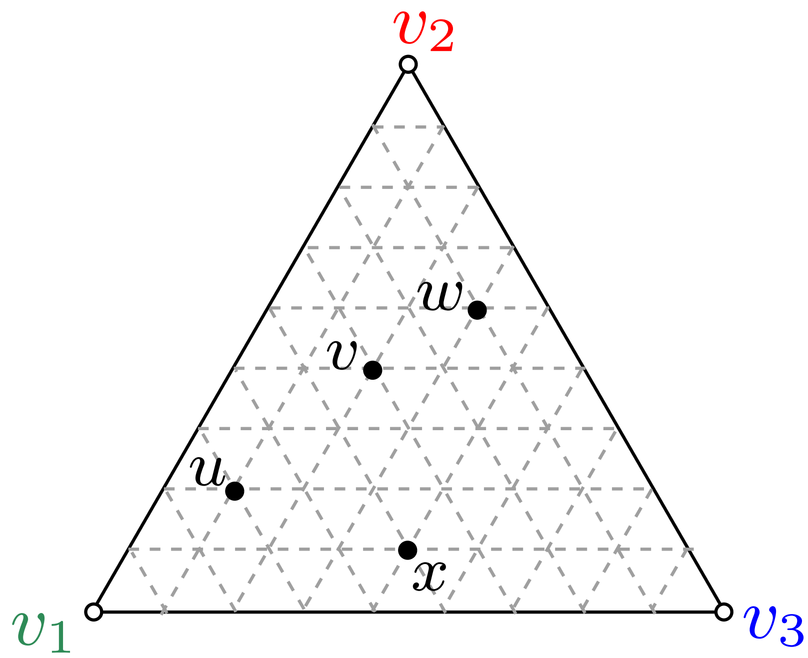




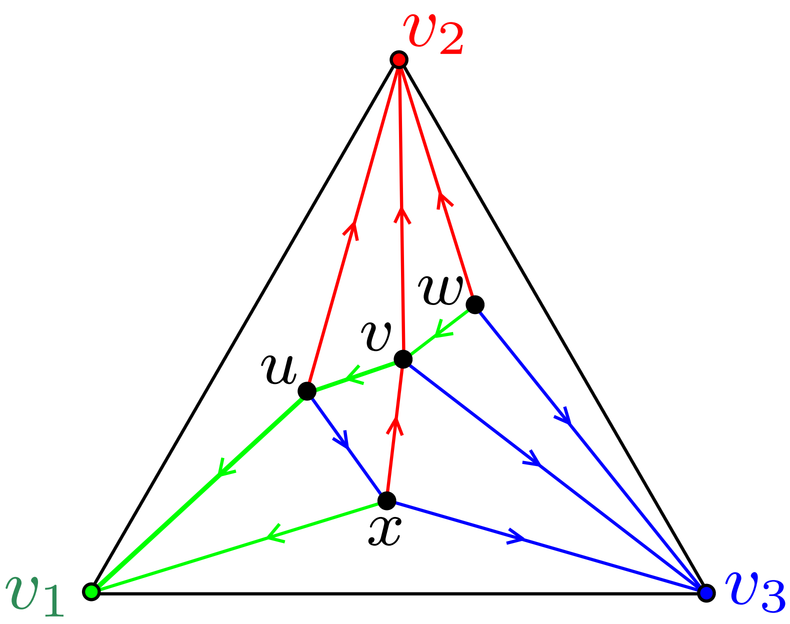
place v at
$\frac{3}{9}v_1 + \frac{4}{9}v_2 + \frac{2}{9}v_3$

Schnyder's face-counting algorithm [Schnyder'90]

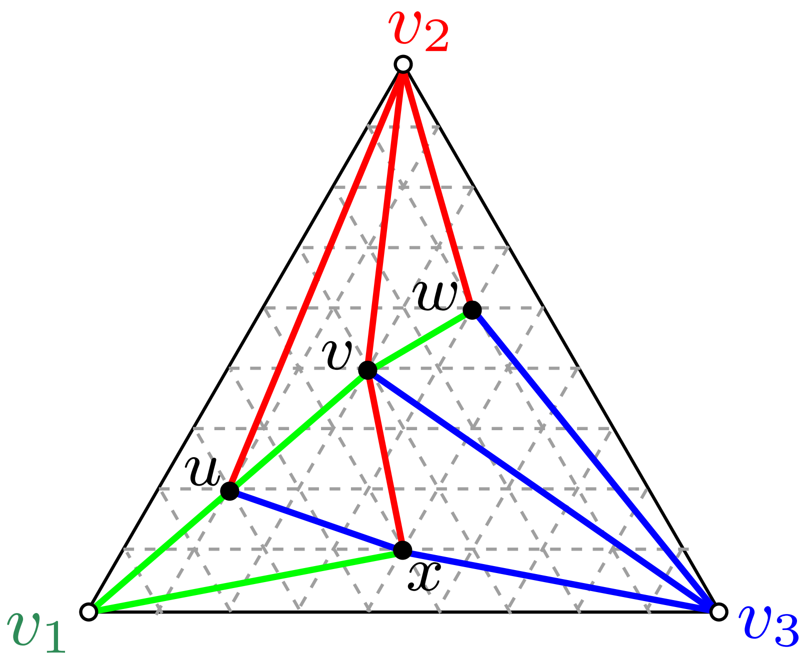




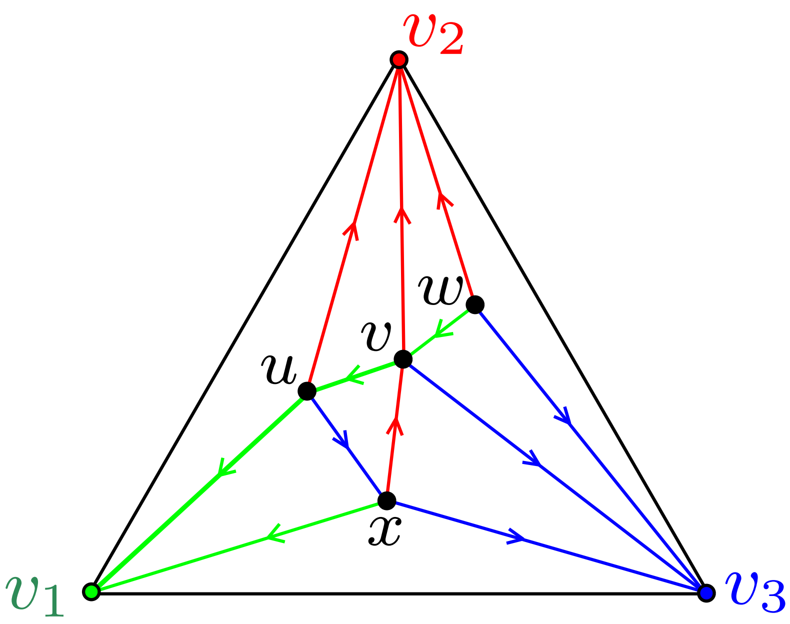
Schnyder's face-counting algorithm [Schnyder'90]



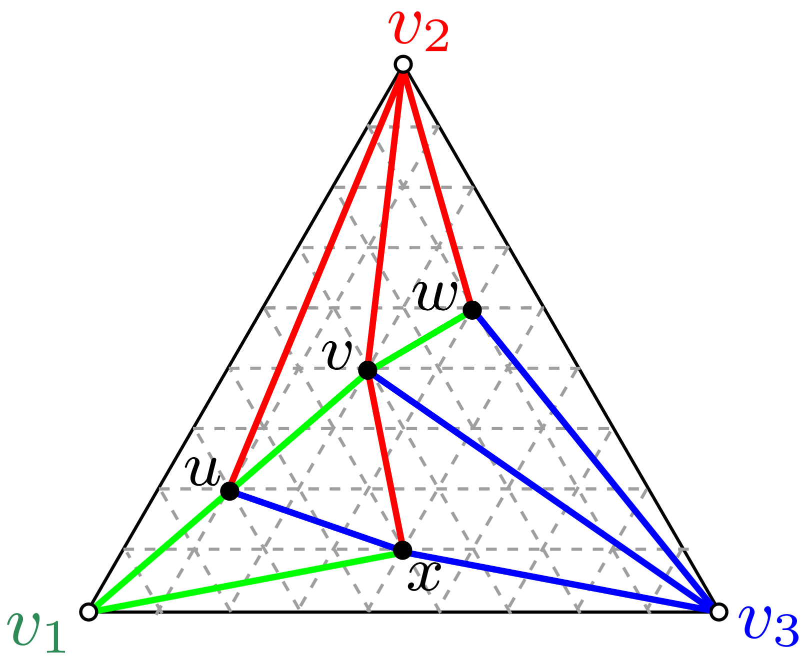
planar straight-line drawing



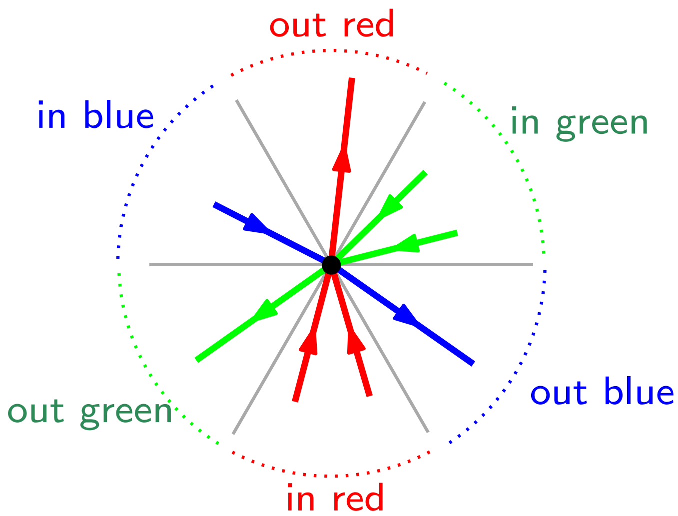
Schnyder's face-counting algorithm [Schnyder'90]



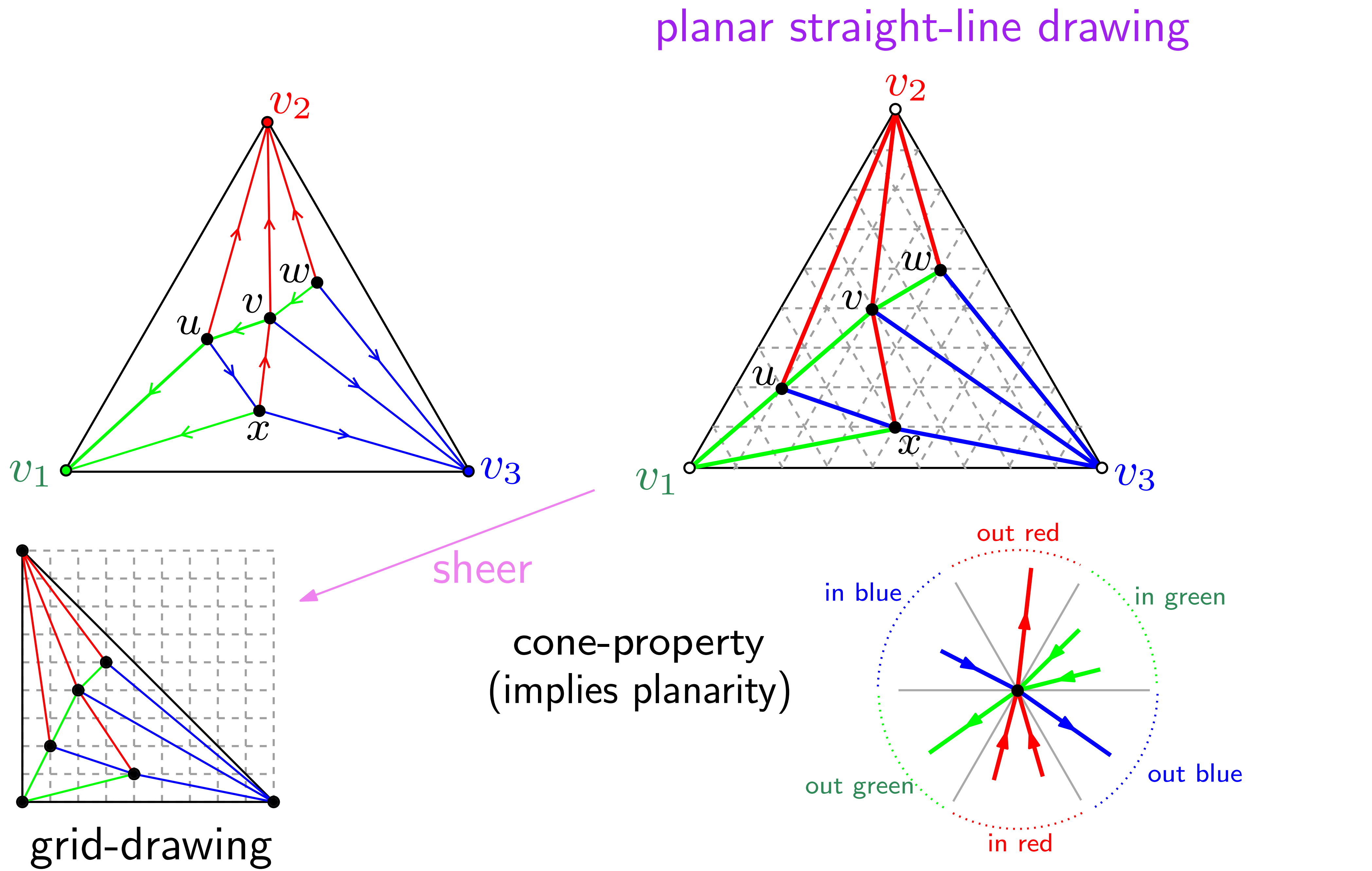
planar straight-line drawing



cone-property
(implies planarity)



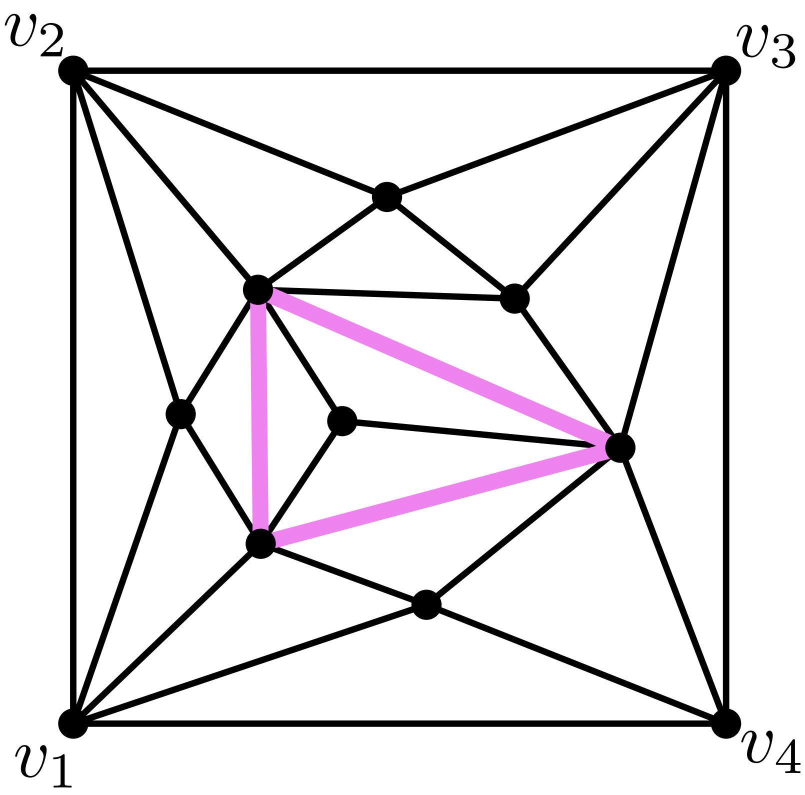
Schnyder's face-counting algorithm [Schnyder'90]



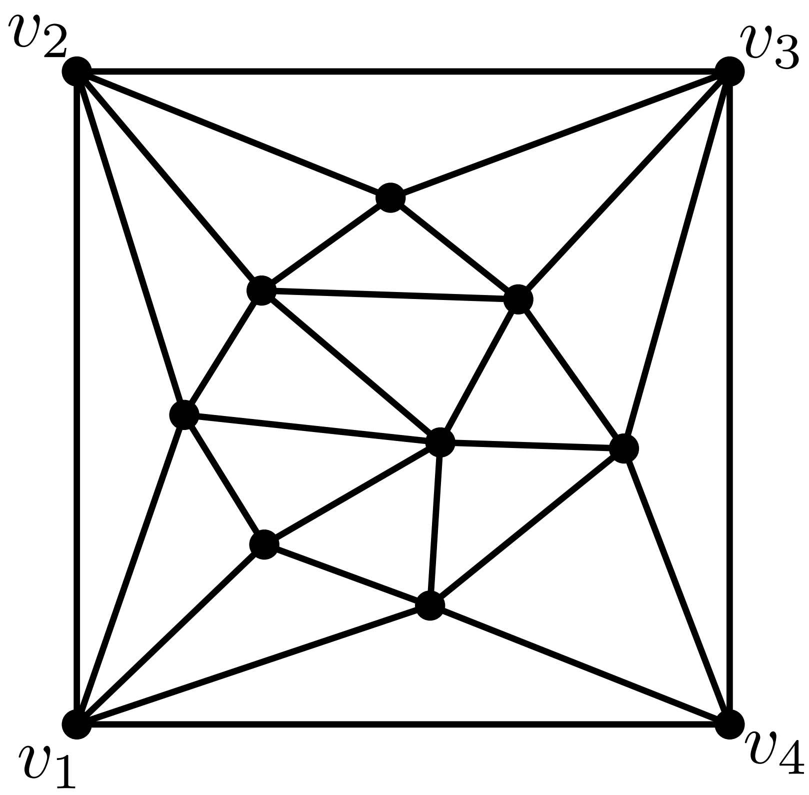
4-connected case

Triangulations of the 4-gon, irreducibility

A triangulation of the 4-gon is **irreducible** if all 3-cycles bound faces



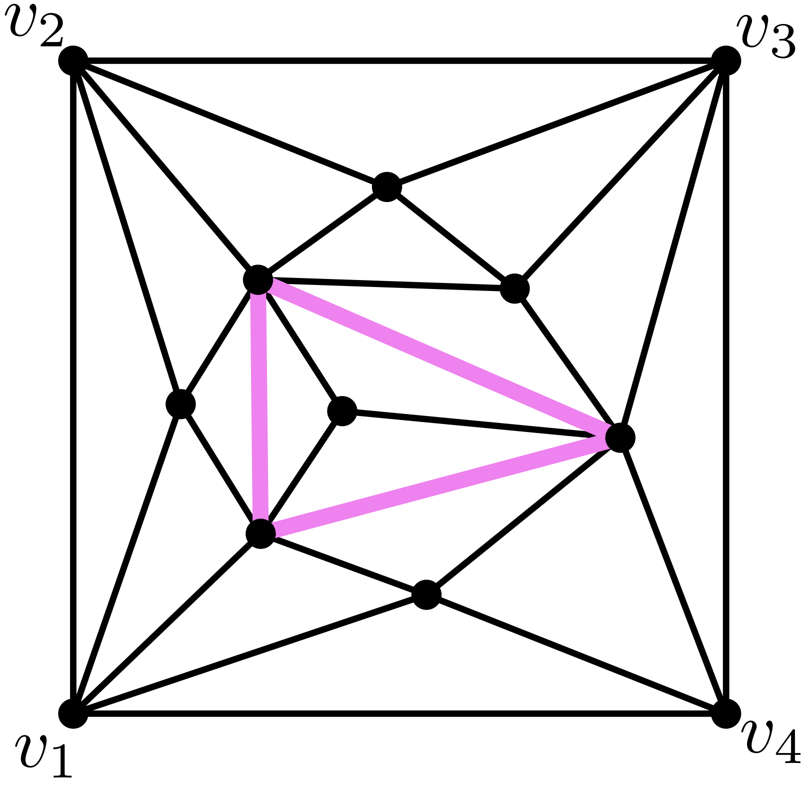
not irreducible



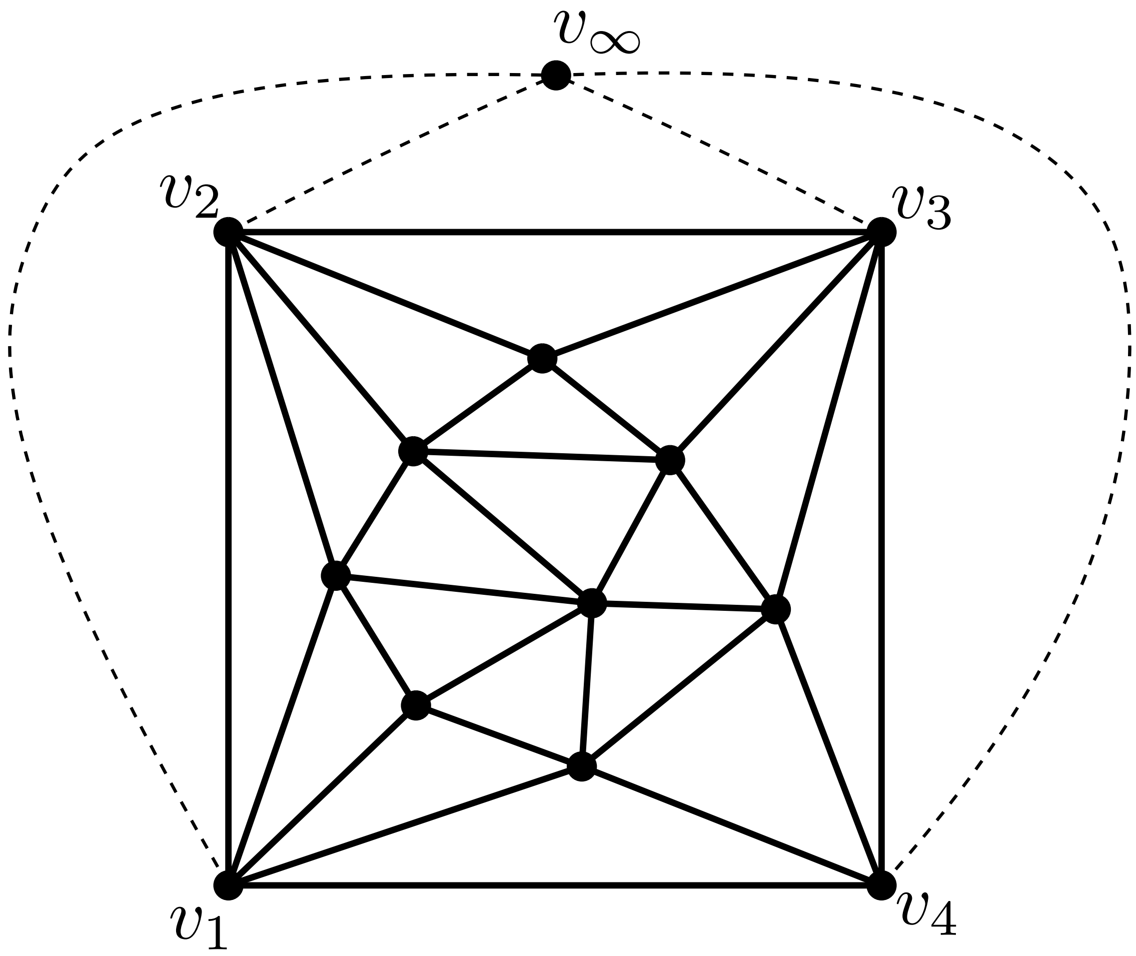
irreducible

Triangulations of the 4-gon, irreducibility

A triangulation of the 4-gon is **irreducible** if all 3-cycles bound faces



not irreducible



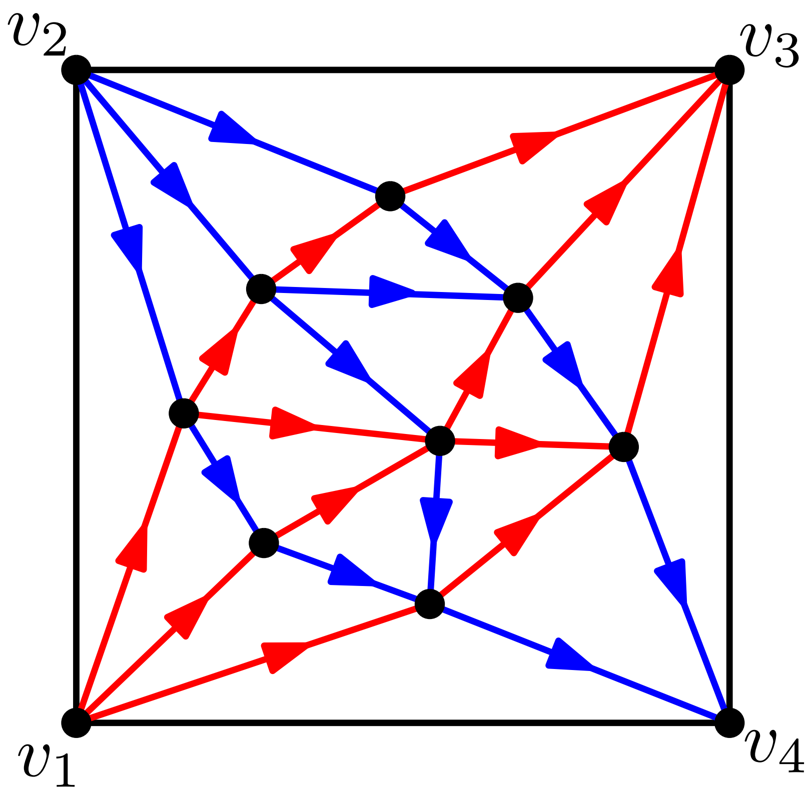
irreducible

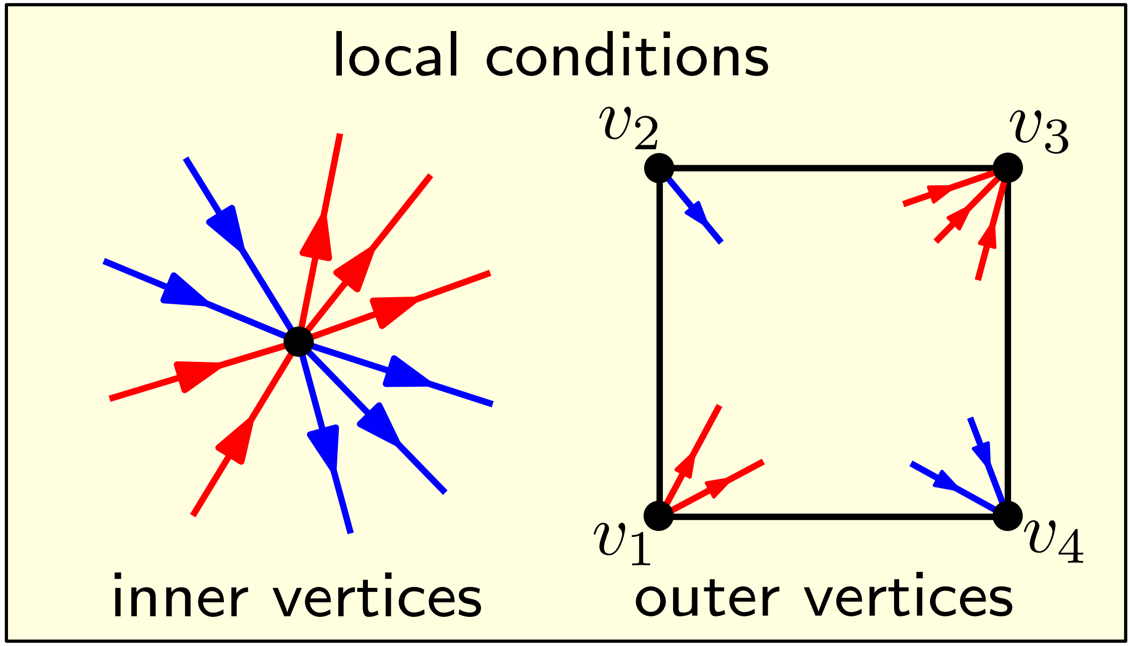
$\Updownarrow$

triangulation augmented by v_∞
is 4-connected

Transversal structures

aka regular edge-labelings [He'93]
(structures dual to rectangular tilings)

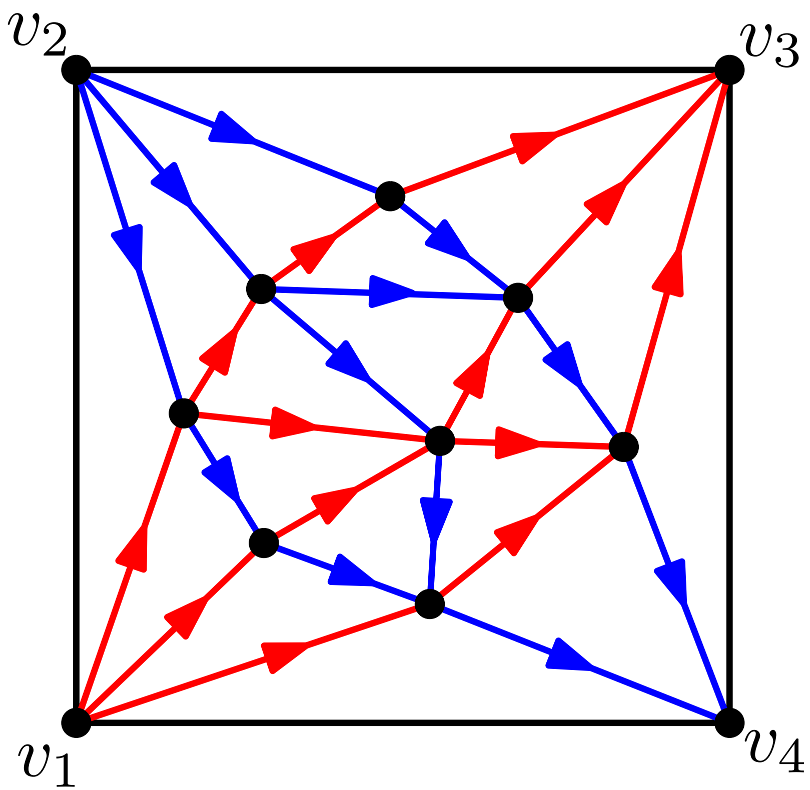


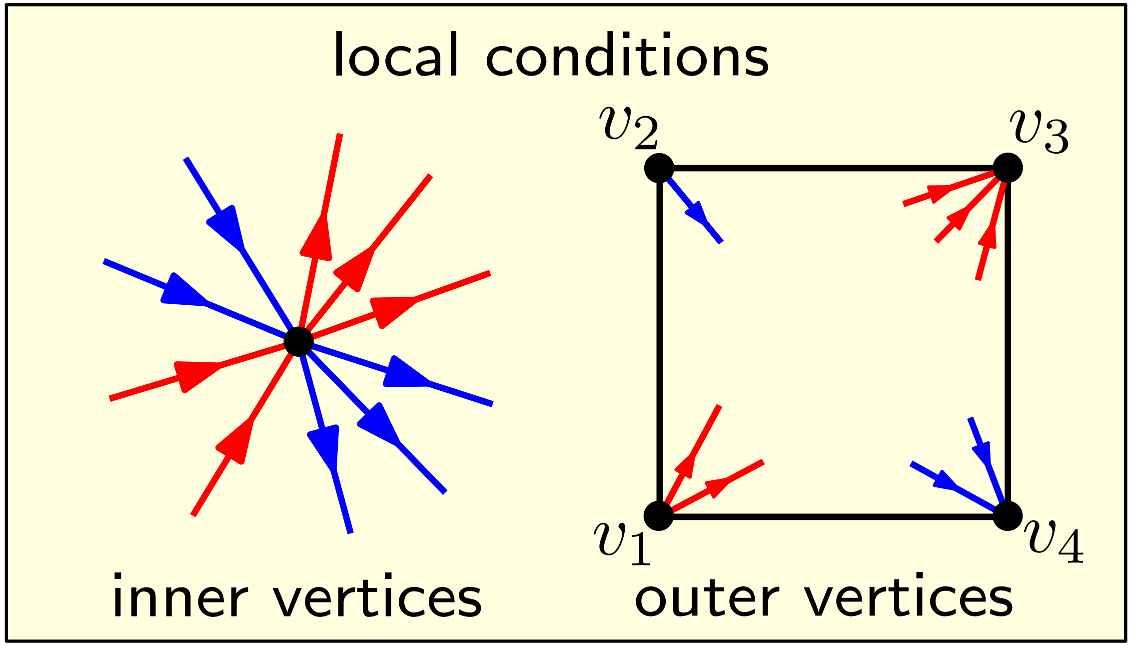


A 4-triangulation admits a transversal structure iff it is irreducible

Transversal structures

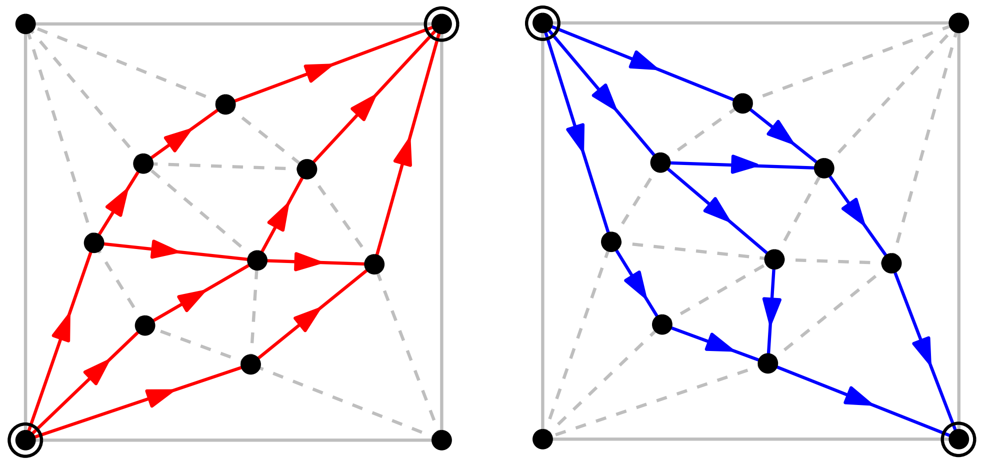
aka regular edge-labelings [He'93]
(structures dual to rectangular tilings)





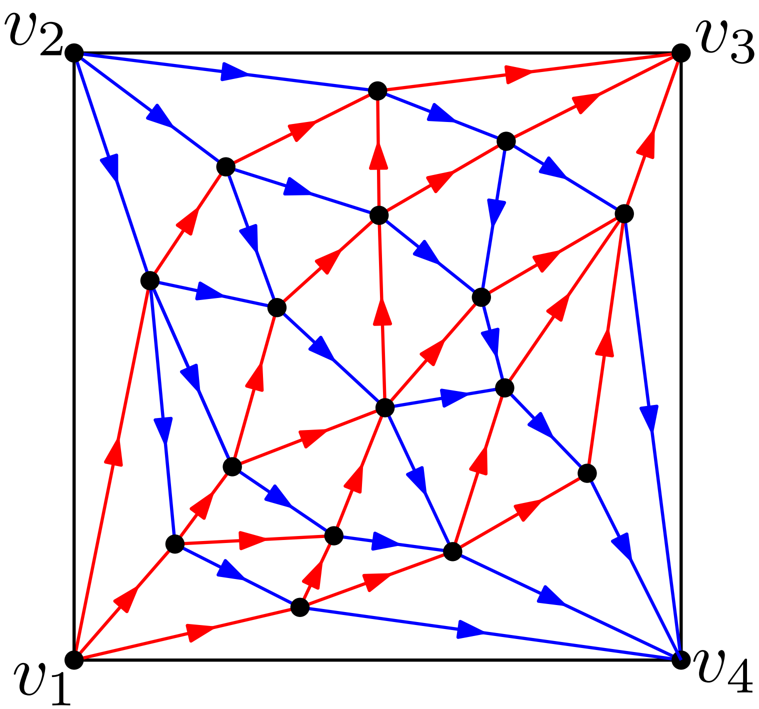
A 4-triangulation admits a transversal structure iff it is irreducible

yields two bipolar orientations:

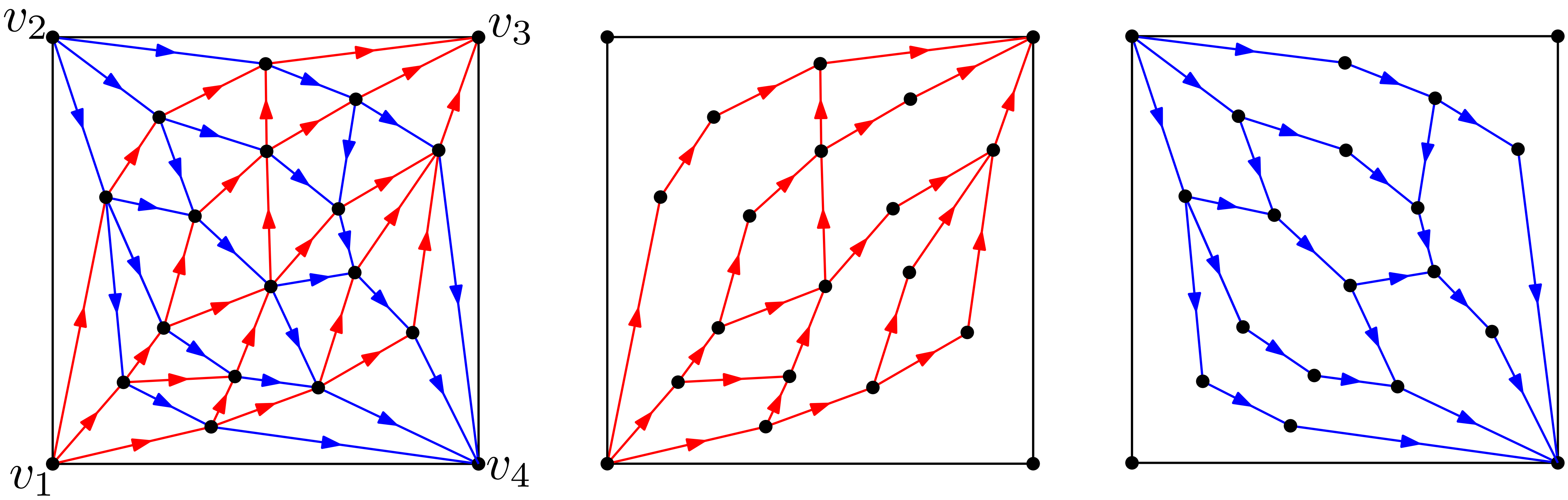


Face-counting algorithm

[F'05]

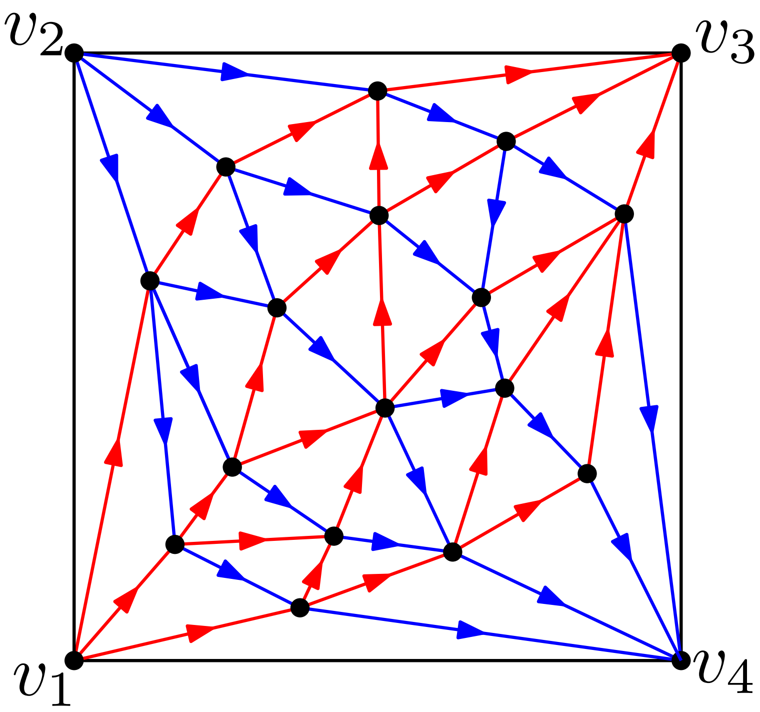


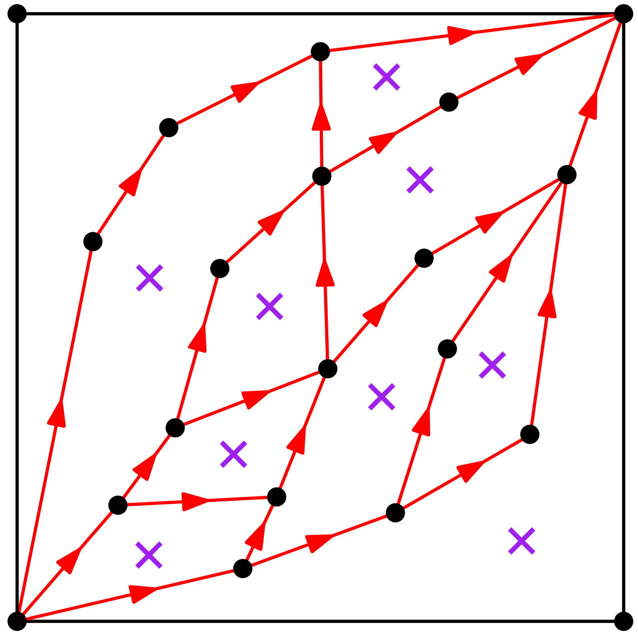
Face-counting algorithm [F'05]



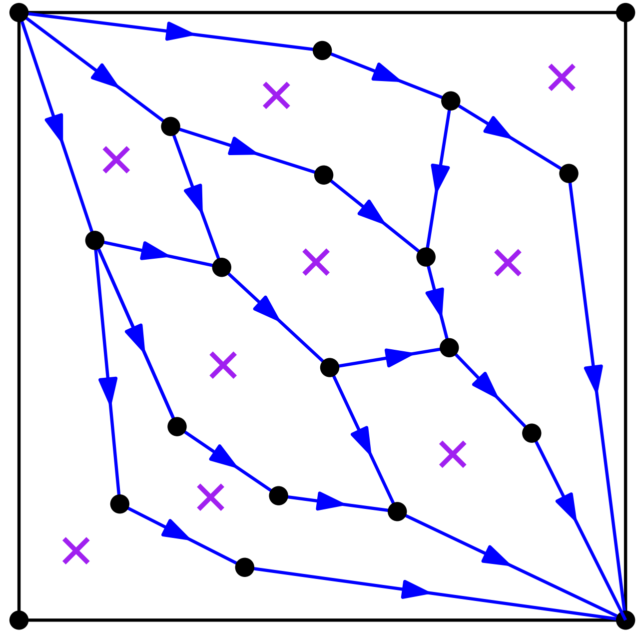
Face-counting algorithm

[F'05]

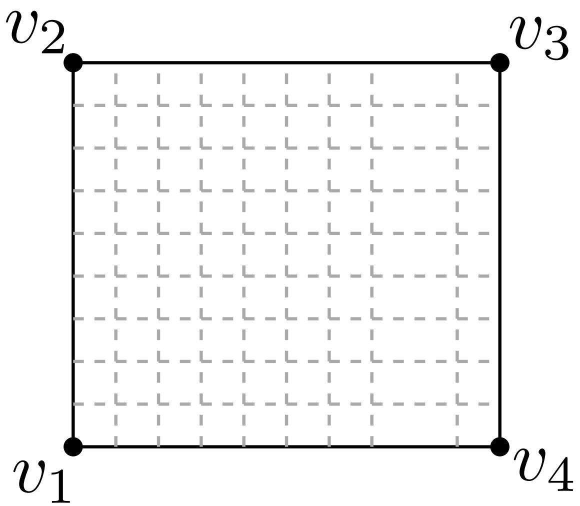




10 faces in red map



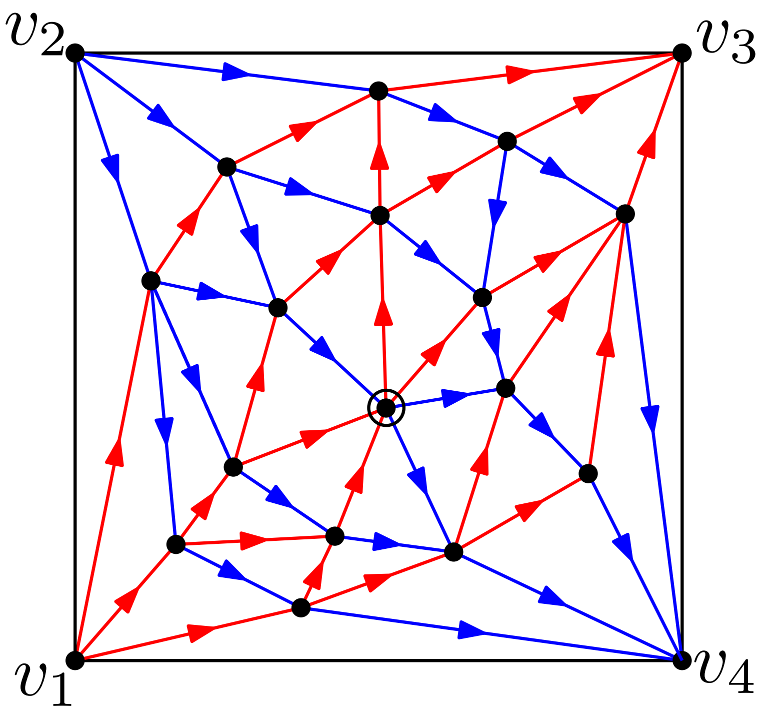
9 faces in blue map

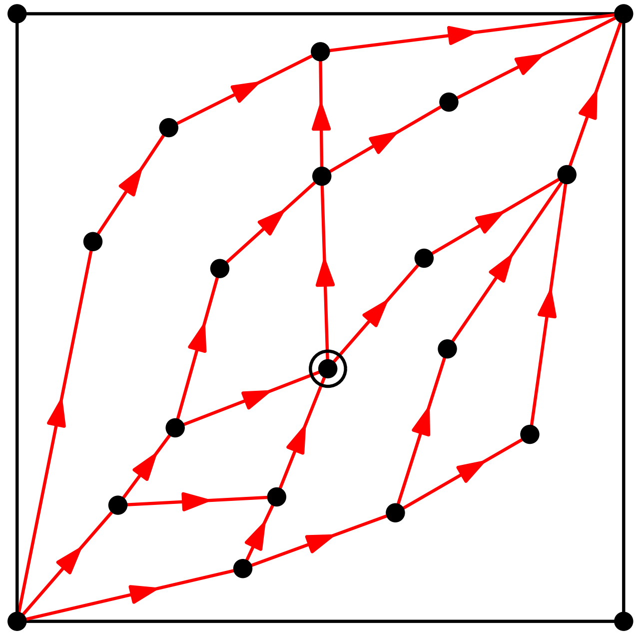


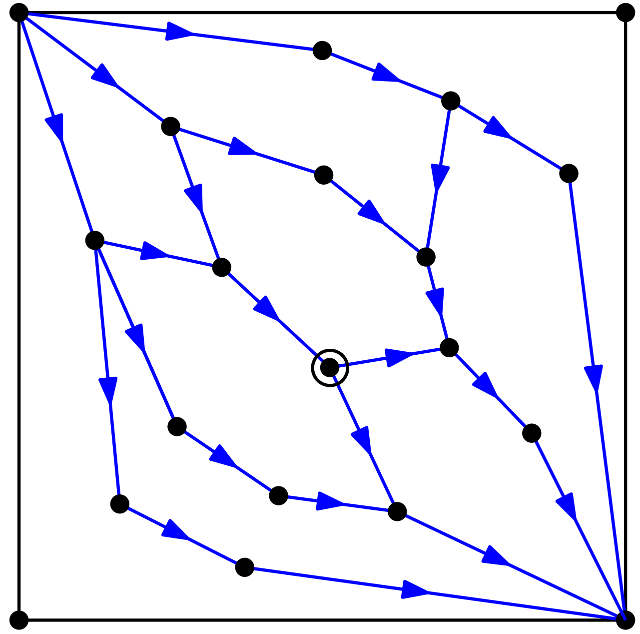
10×9 grid

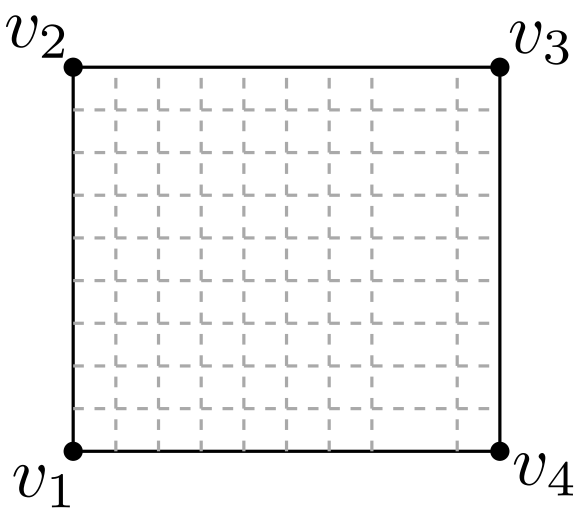
Face-counting algorithm

[F'05]



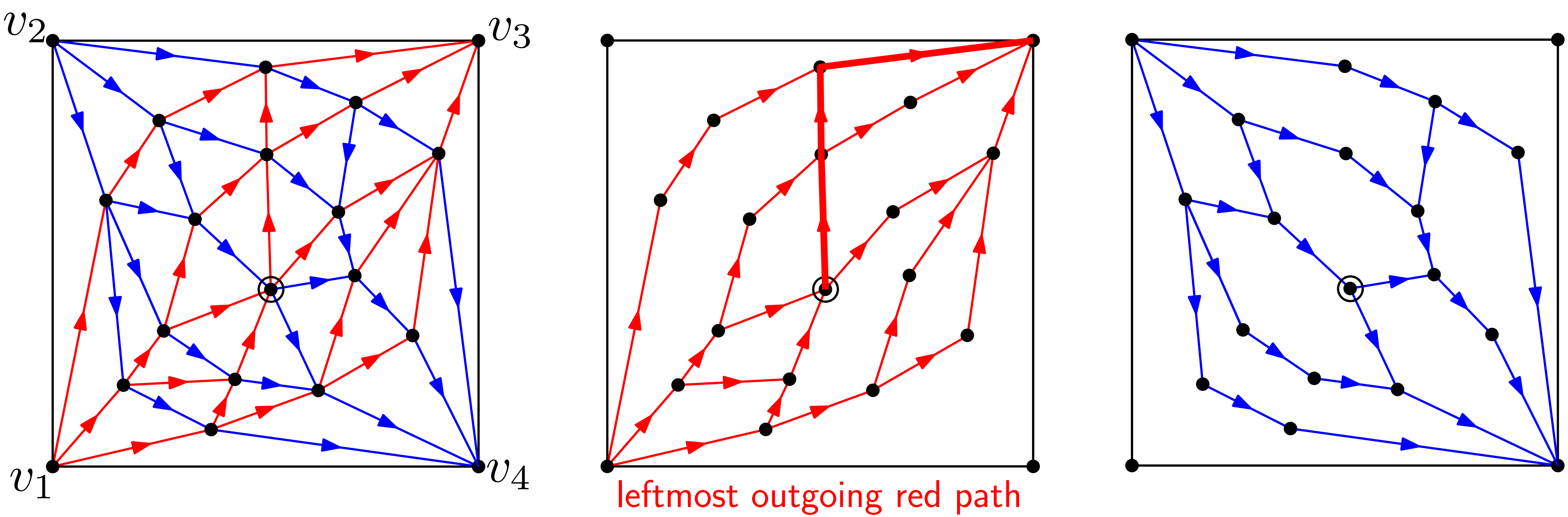


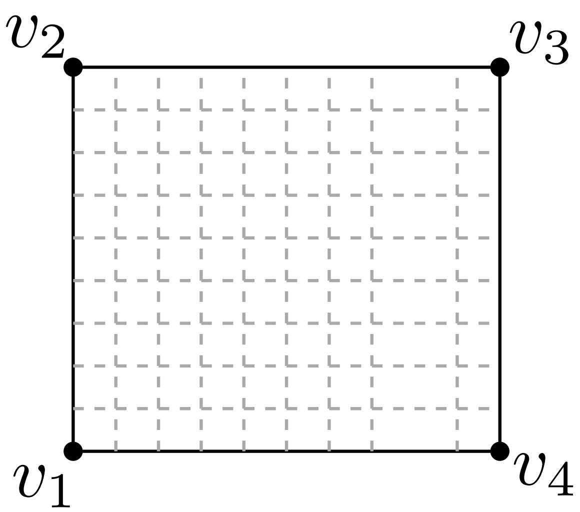




Face-counting algorithm

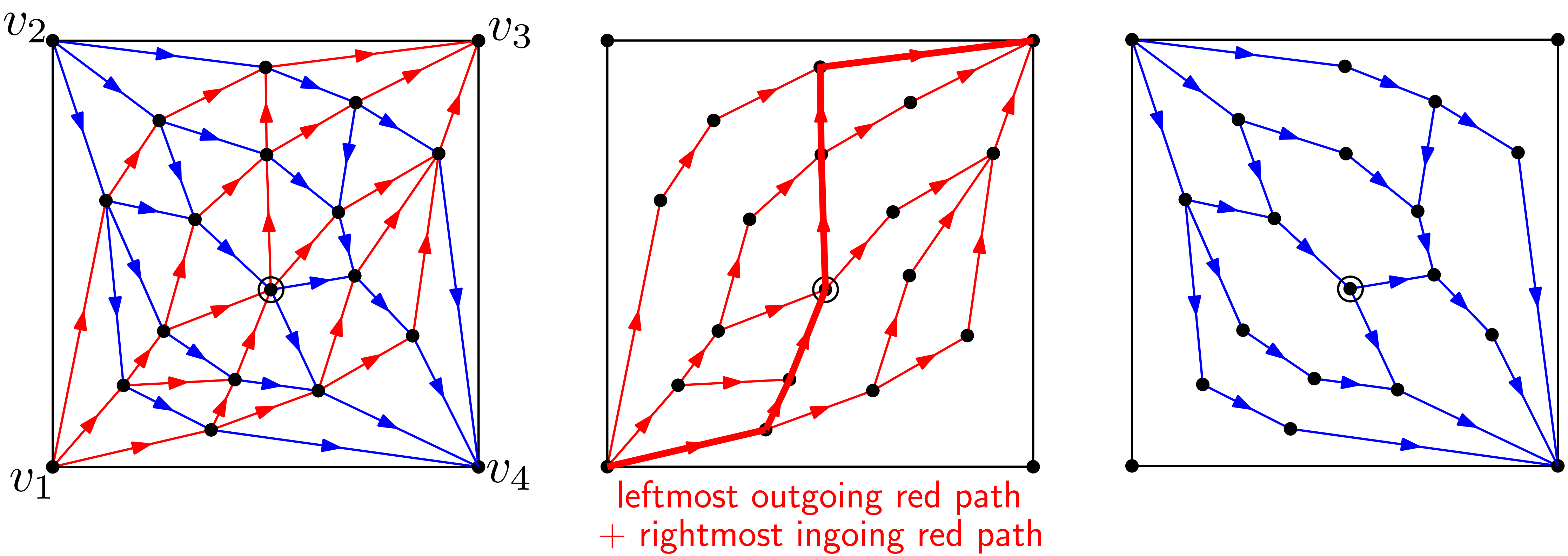
[F'05]

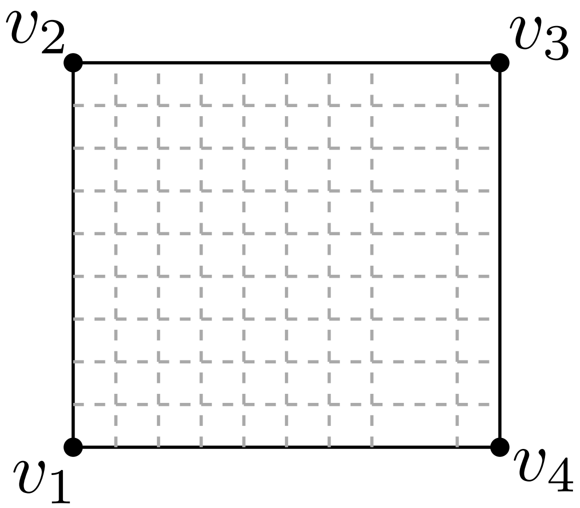




Face-counting algorithm

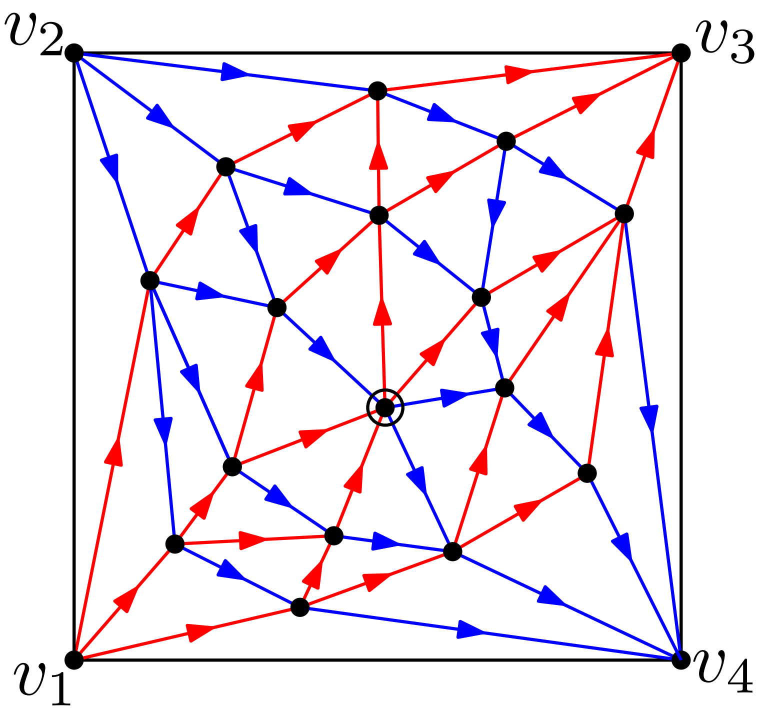
[F'05]

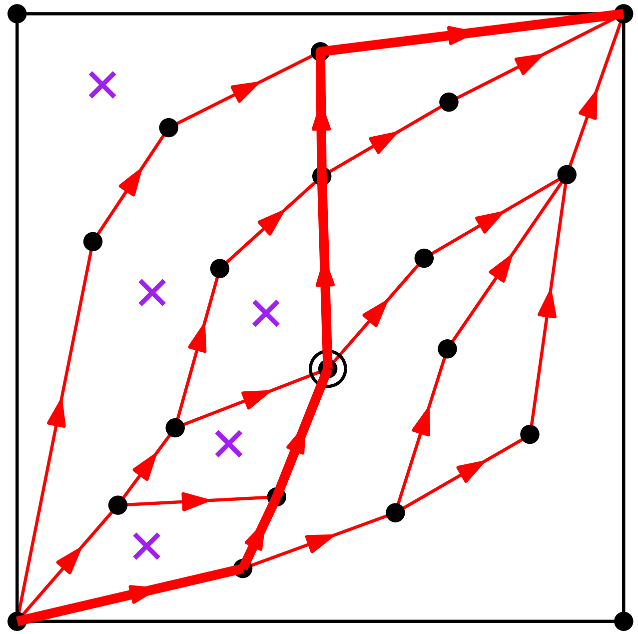




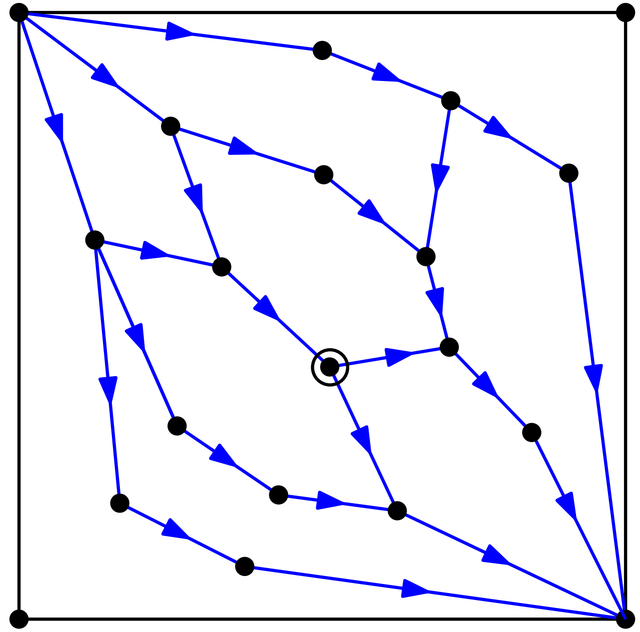
Face-counting algorithm

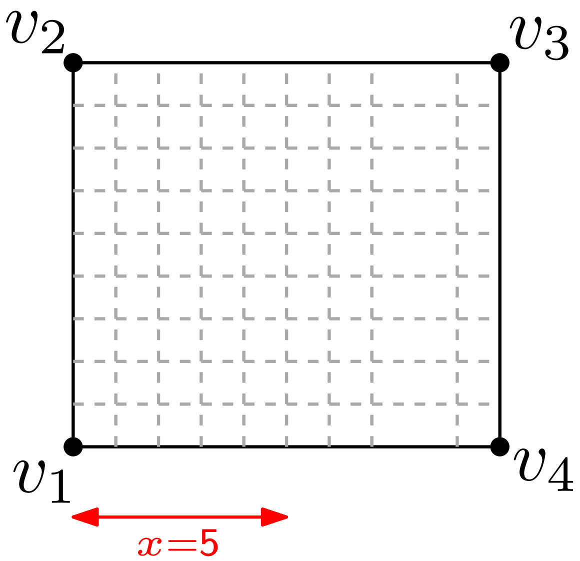
[F'05]





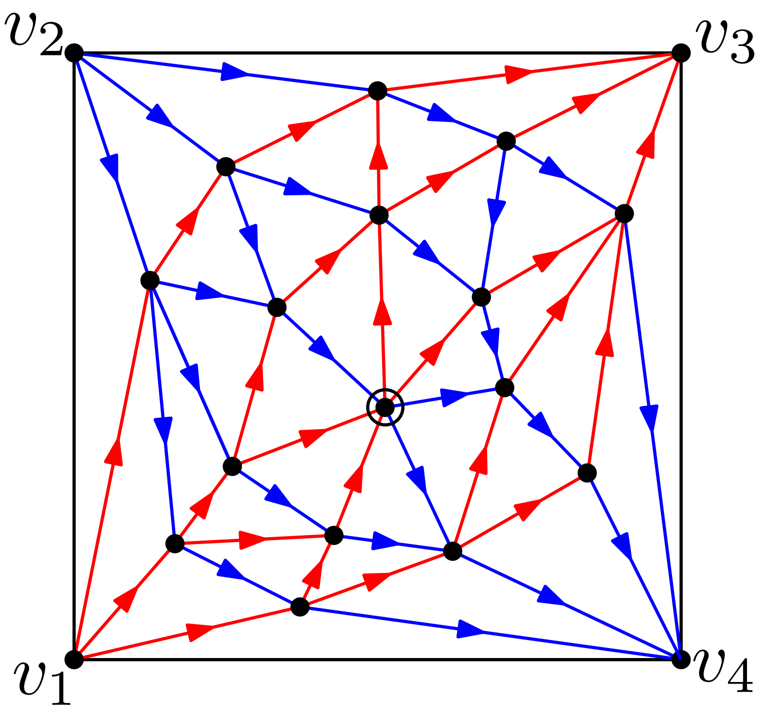
leftmost outgoing red path
+ rightmost ingoing red path

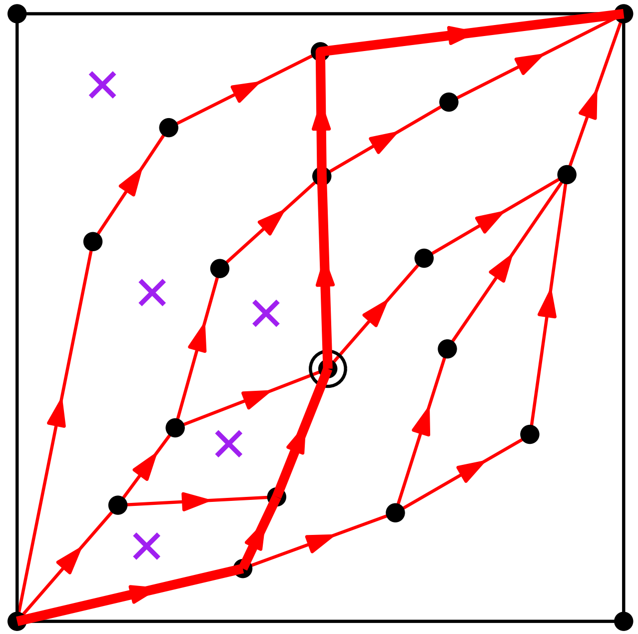




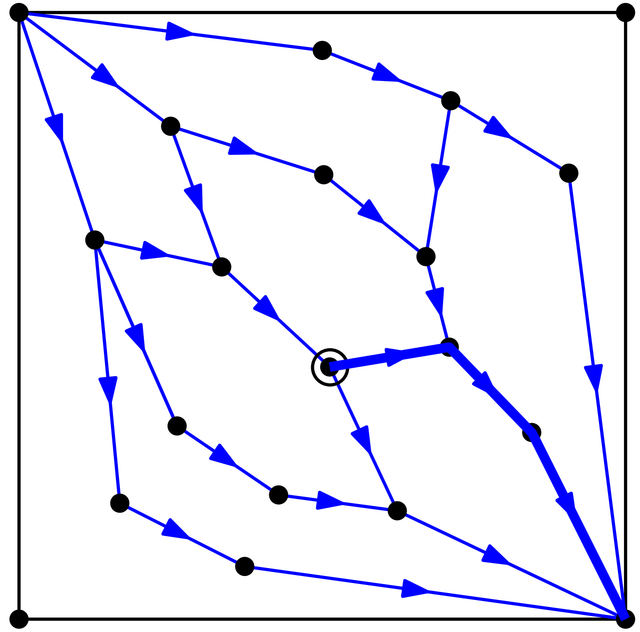
Face-counting algorithm

[F'05]

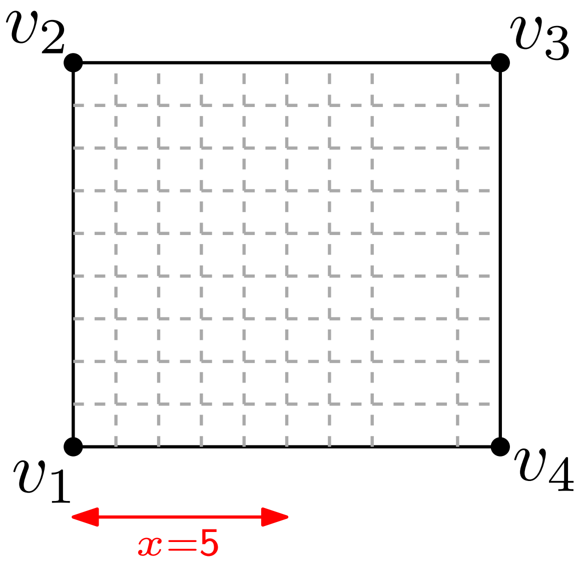




leftmost outgoing red path
+ rightmost ingoing red path

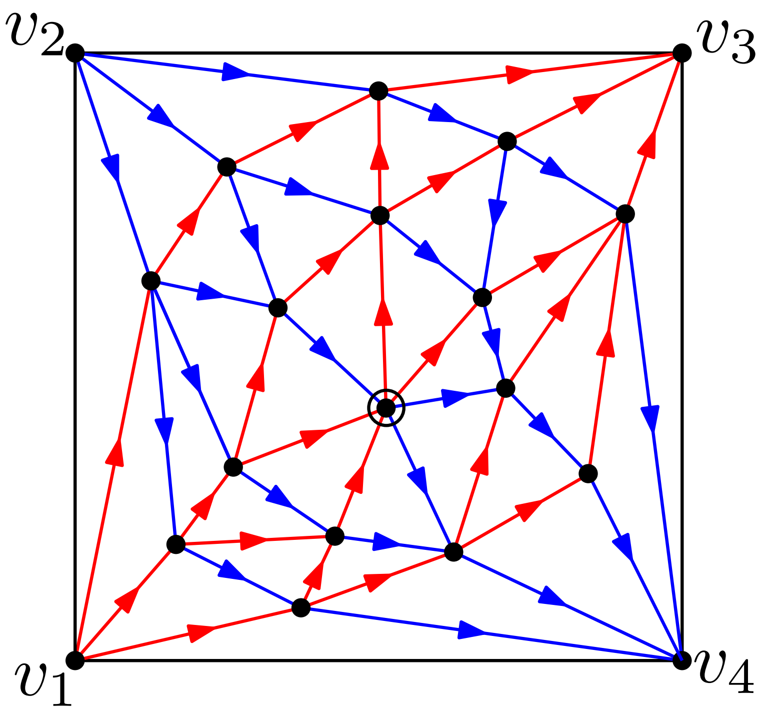


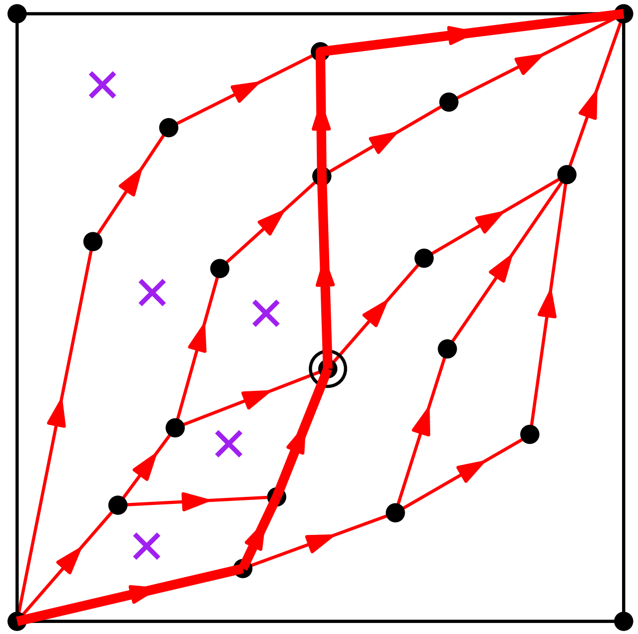
leftmost outgoing blue path



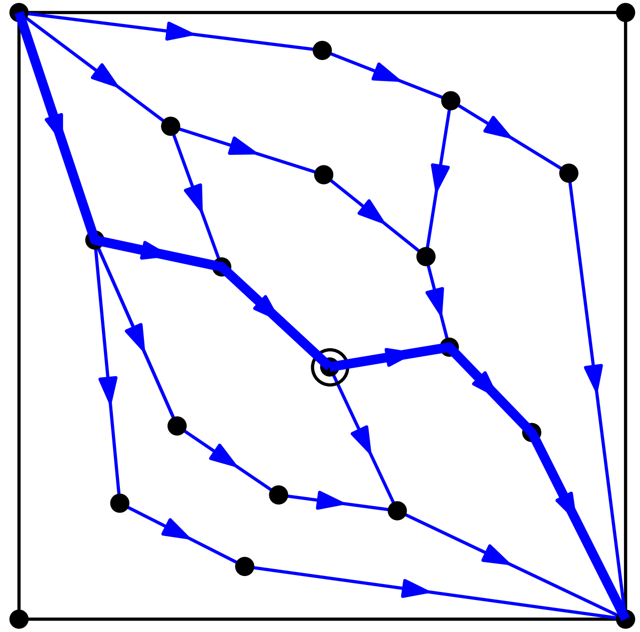
Face-counting algorithm

[F'05]

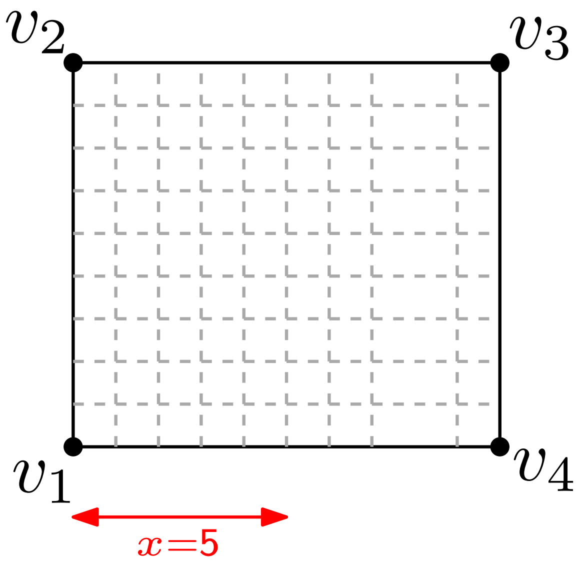




leftmost outgoing red path
+ rightmost ingoing red path

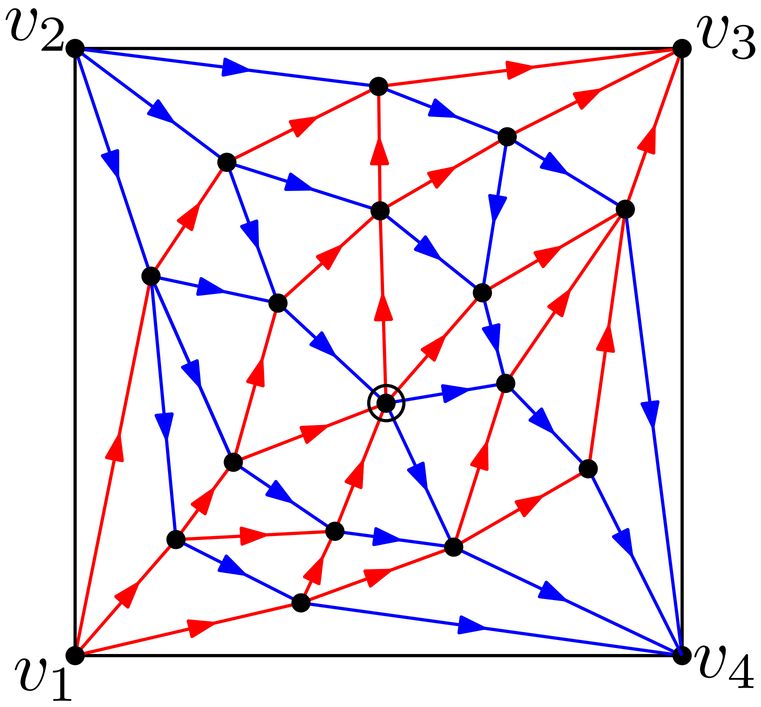


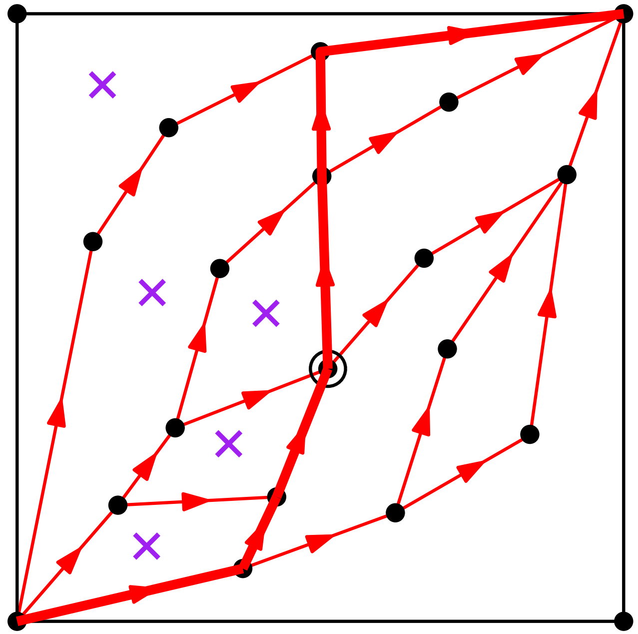
leftmost outgoing blue path
+ rightmost ingoing blue path



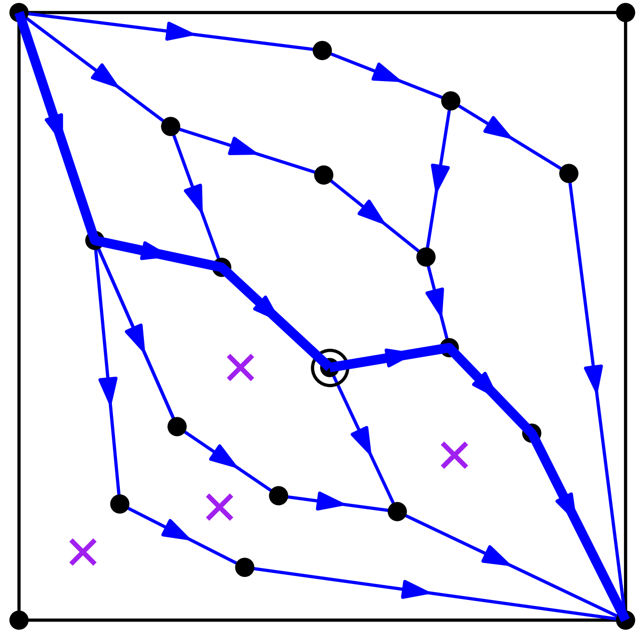
Face-counting algorithm

[F'05]

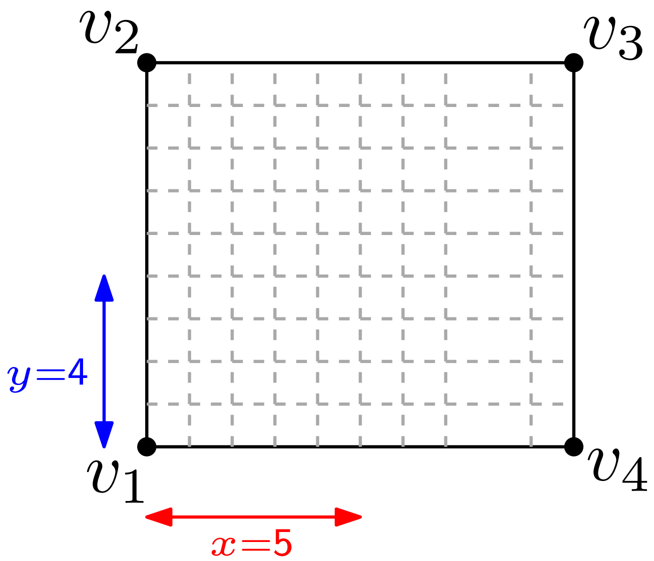




leftmost outgoing red path
+ rightmost ingoing red path

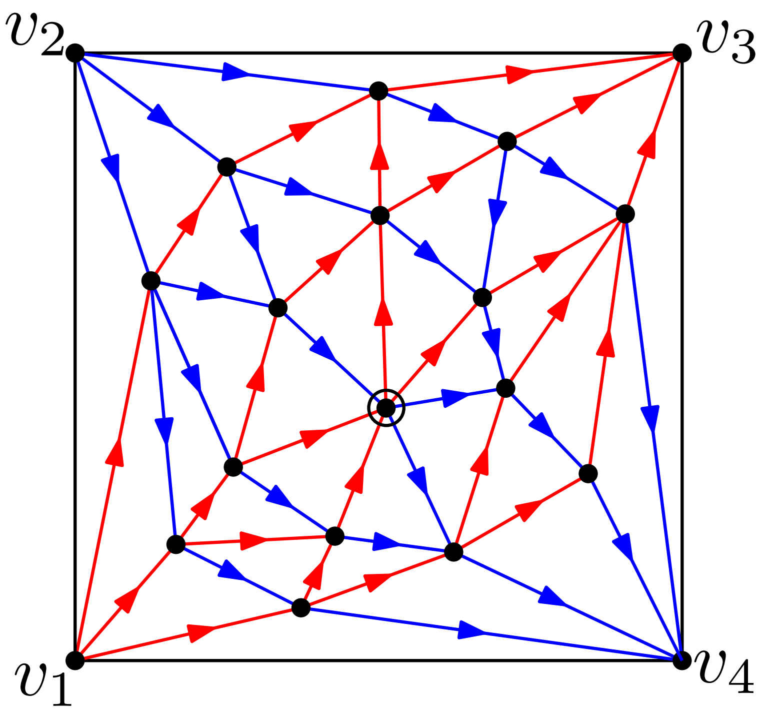


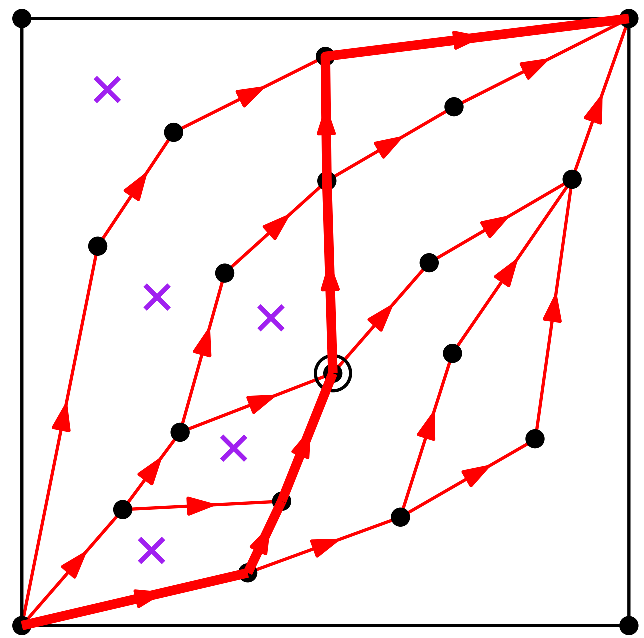
leftmost outgoing blue path
+ rightmost ingoing blue path



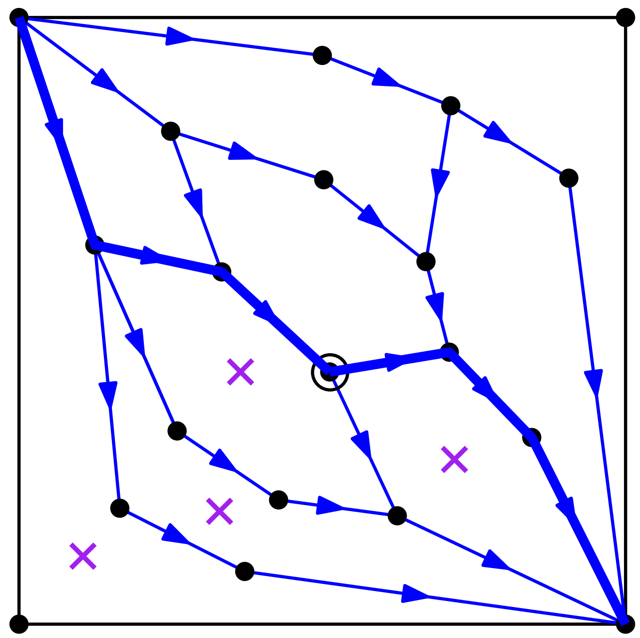
Face-counting algorithm

[F'05]

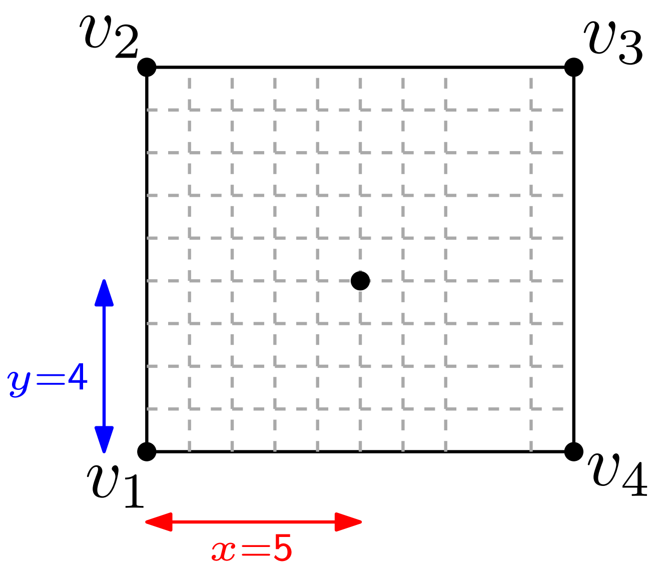




leftmost outgoing red path
+ rightmost ingoing red path

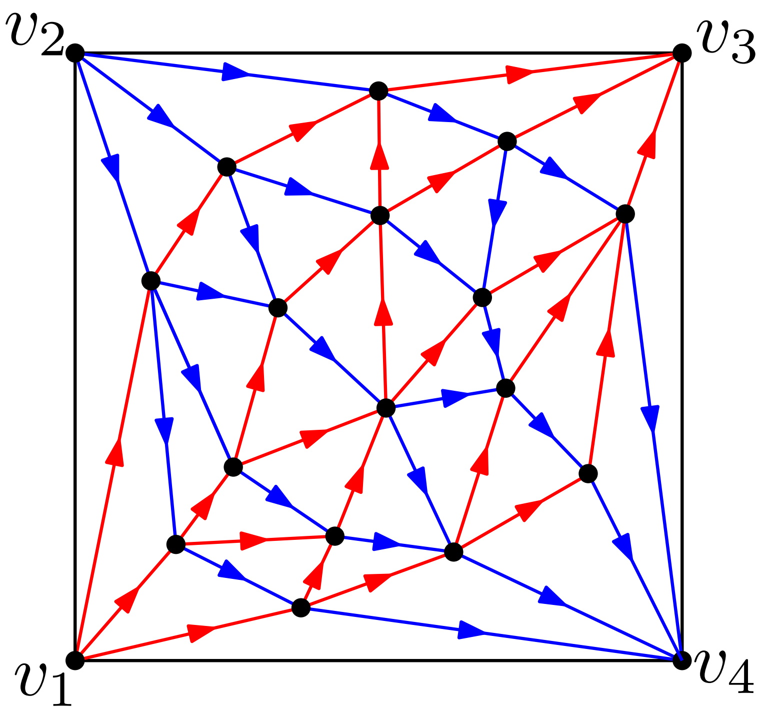


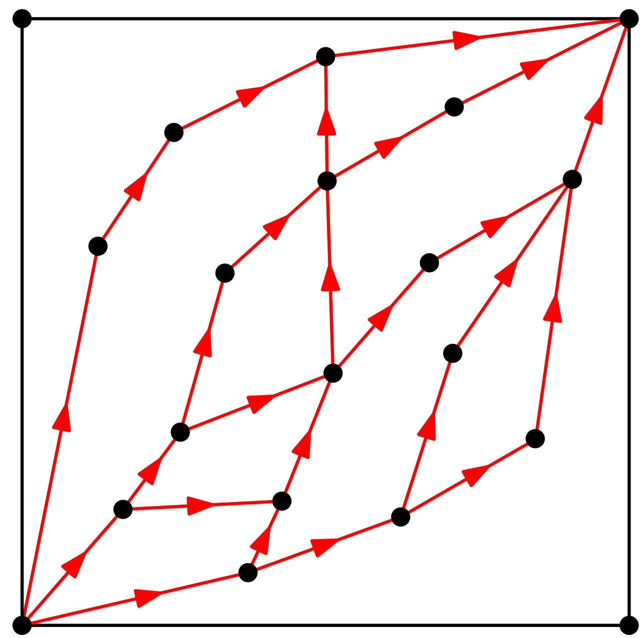
leftmost outgoing blue path
+ rightmost ingoing blue path

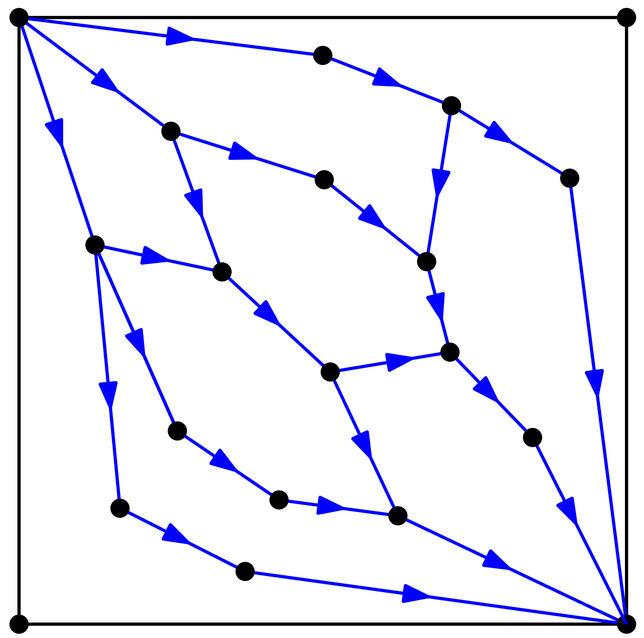


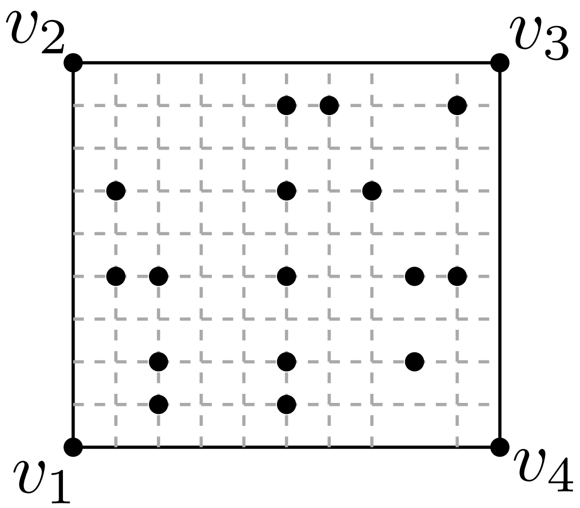
Face-counting algorithm

[F'05]



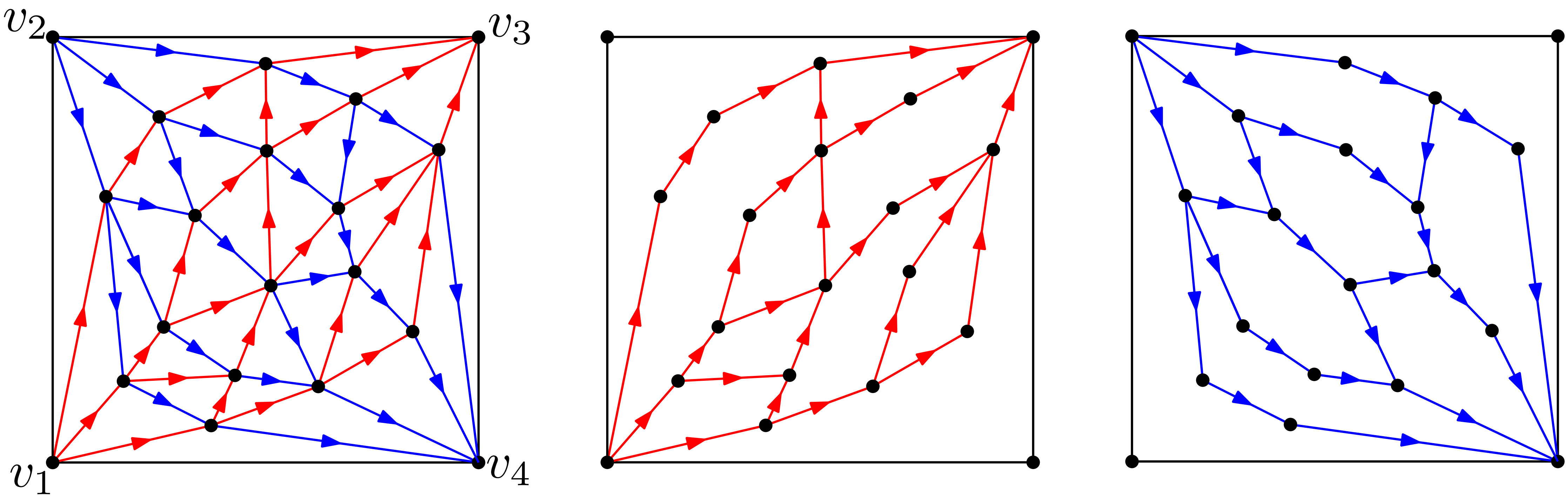


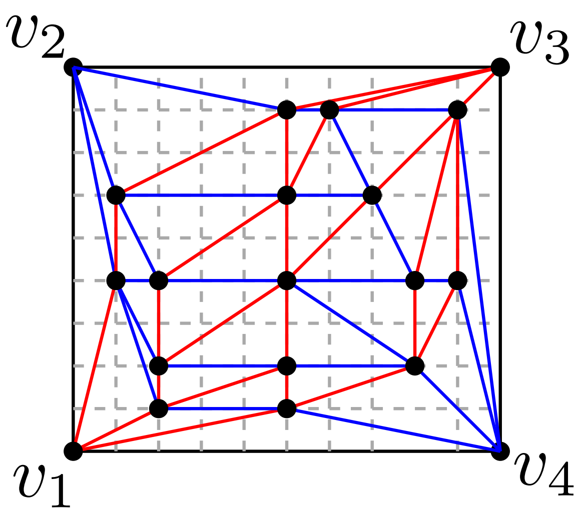




Face-counting algorithm

[F'05]

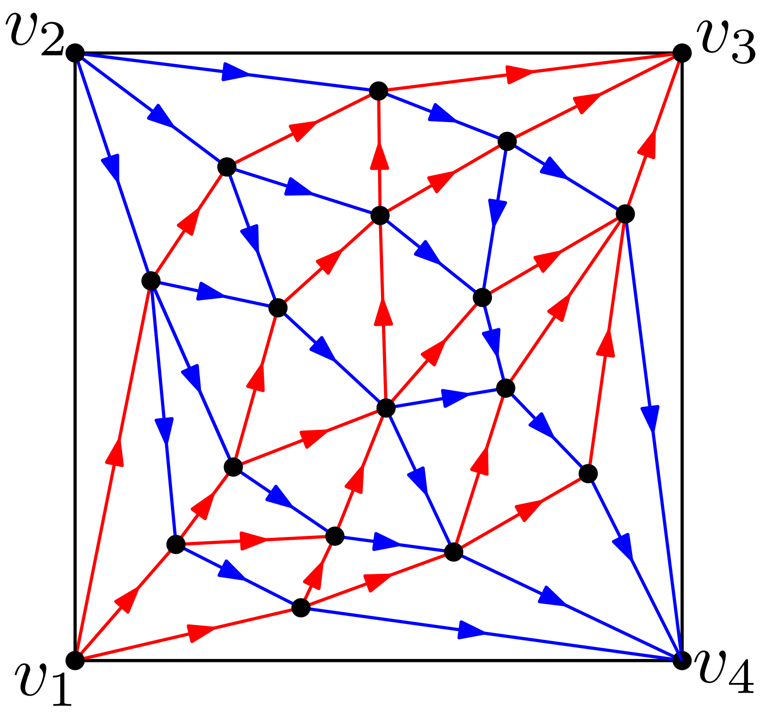


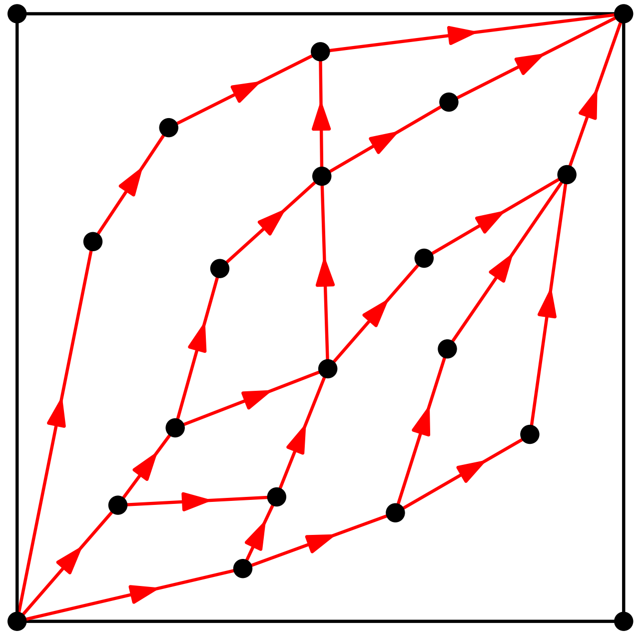


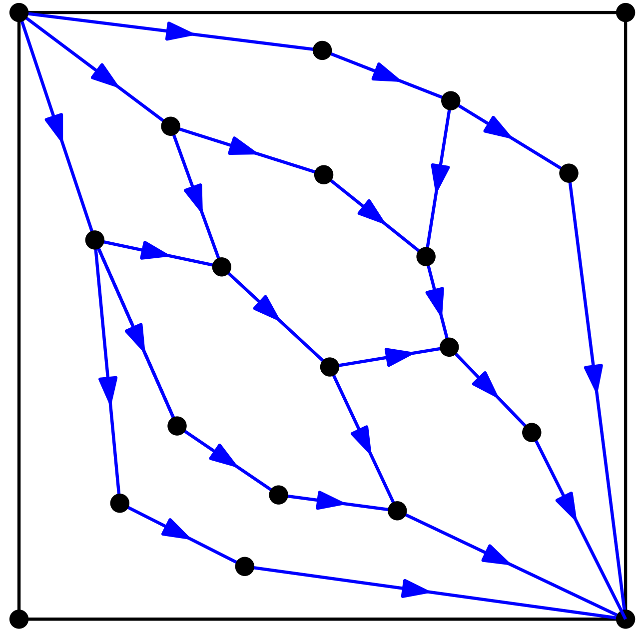
planar straight-line drawing

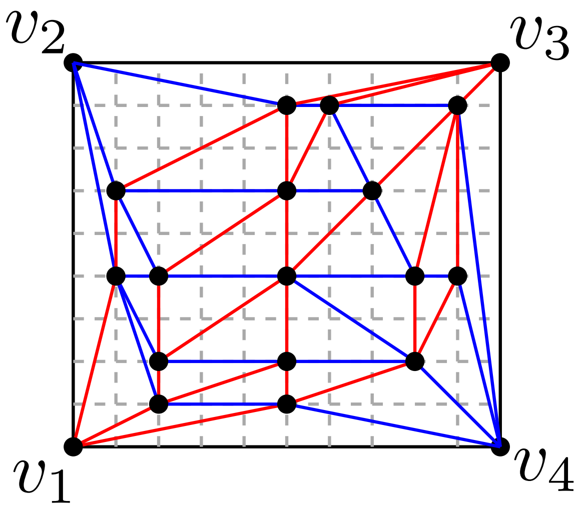
Face-counting algorithm

[F'05]



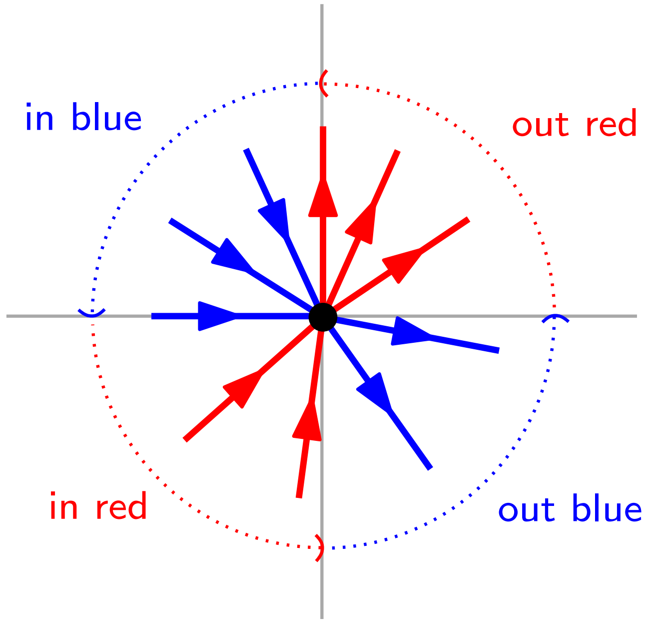




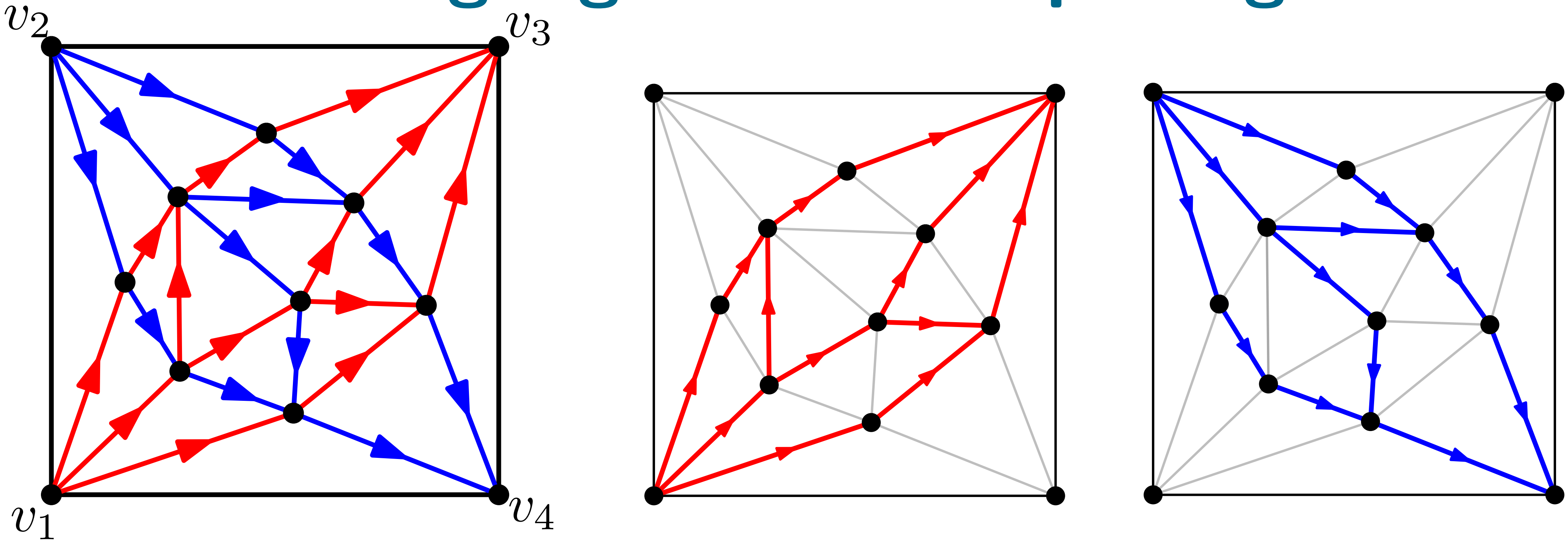


planar straight-line drawing

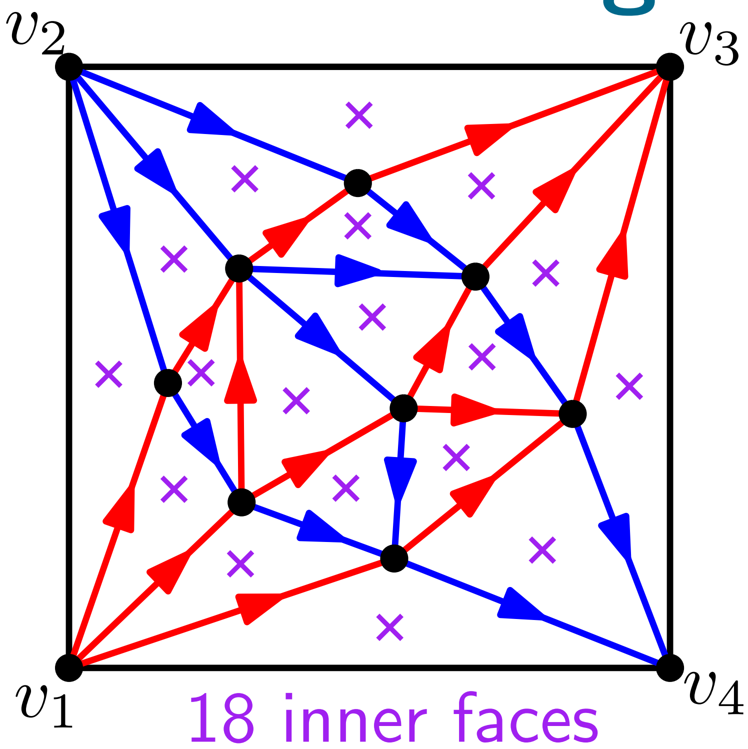
cone property (implies planarity)

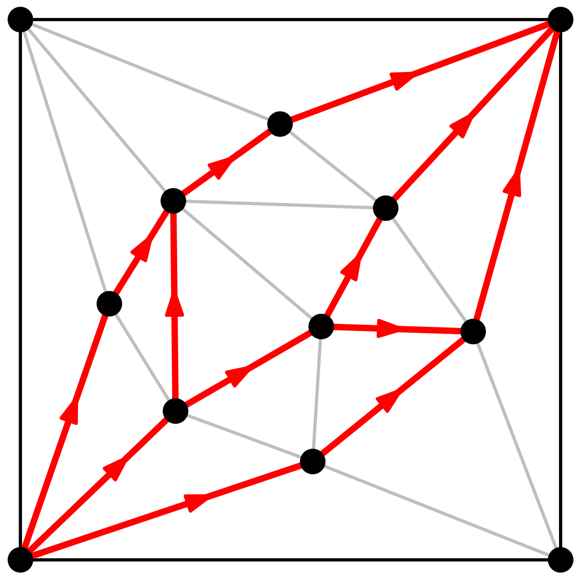


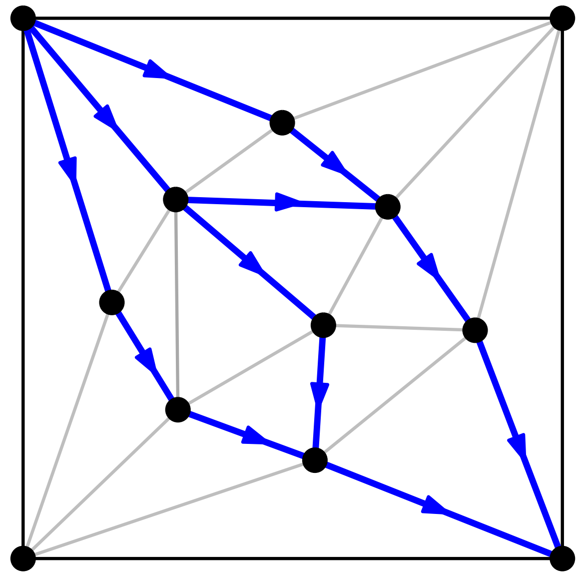
Face-counting algorithm on square grid

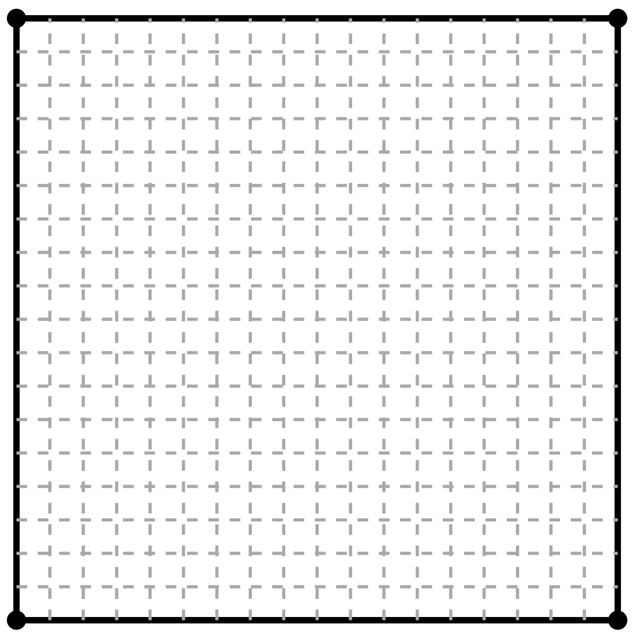


Face-counting algorithm on square grid



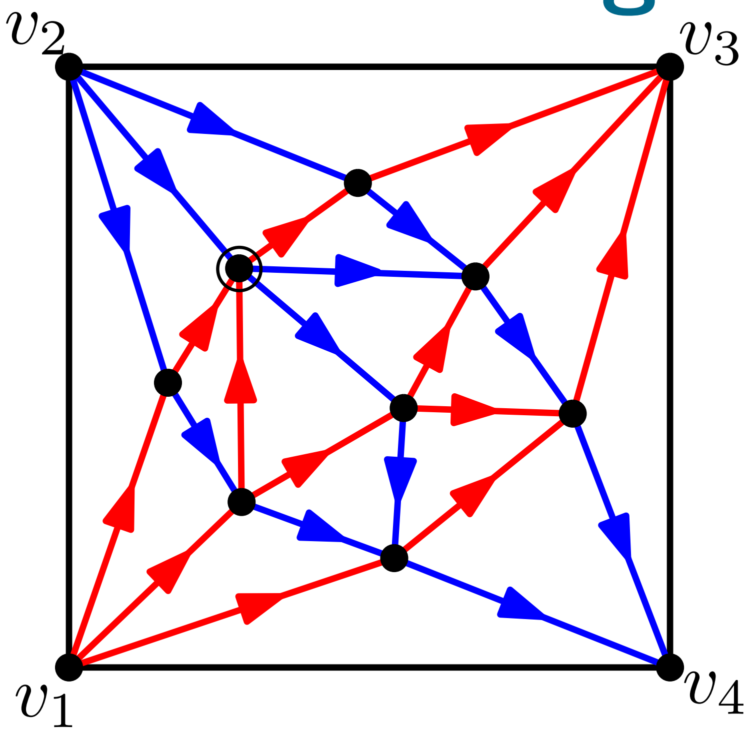


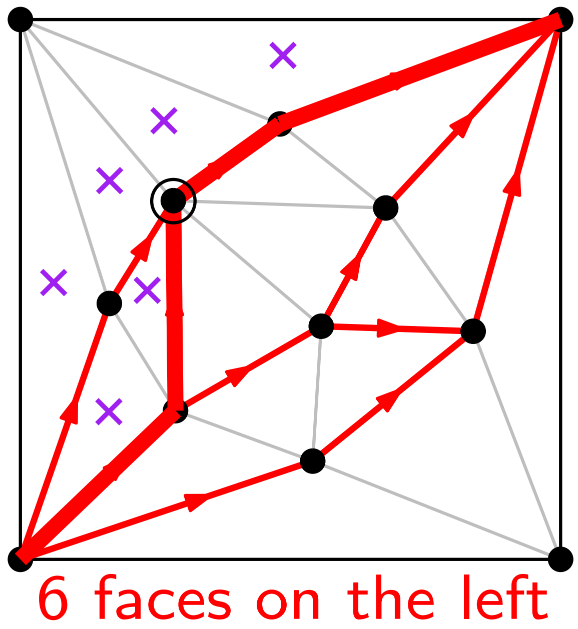


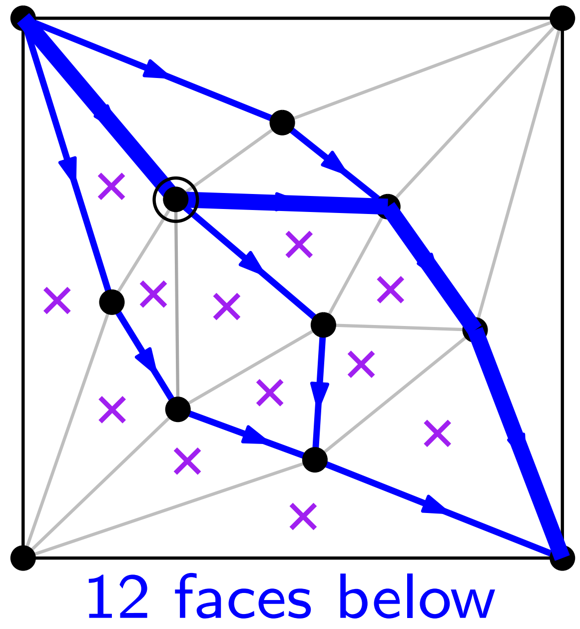


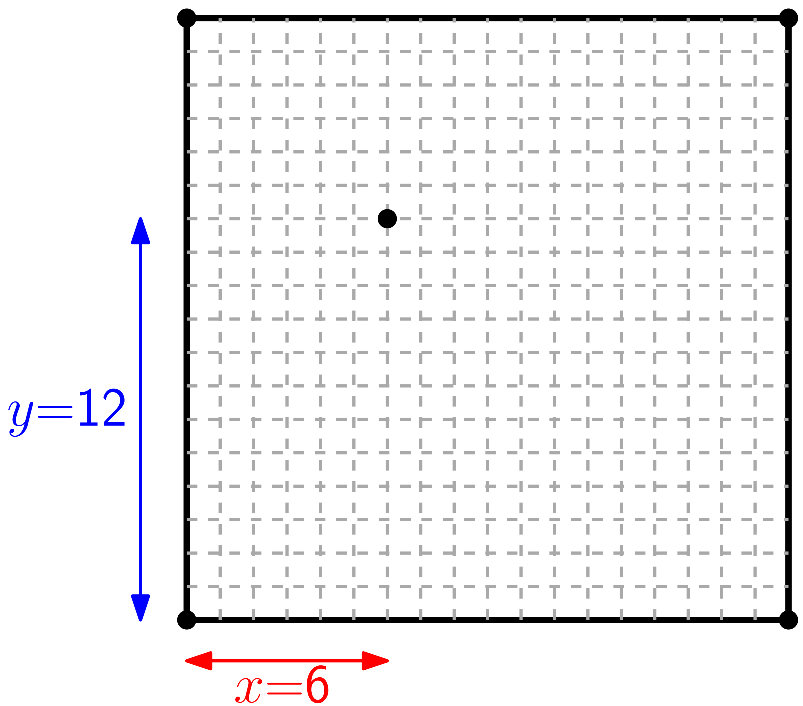
18×18 grid

Face-counting algorithm on square grid

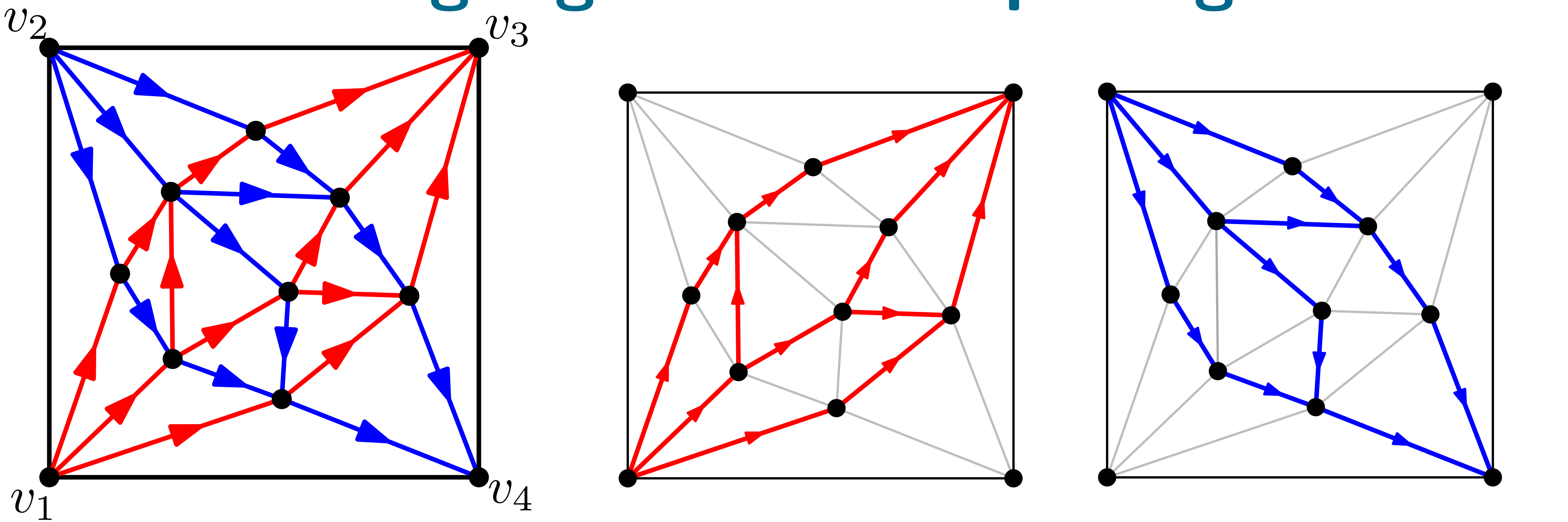


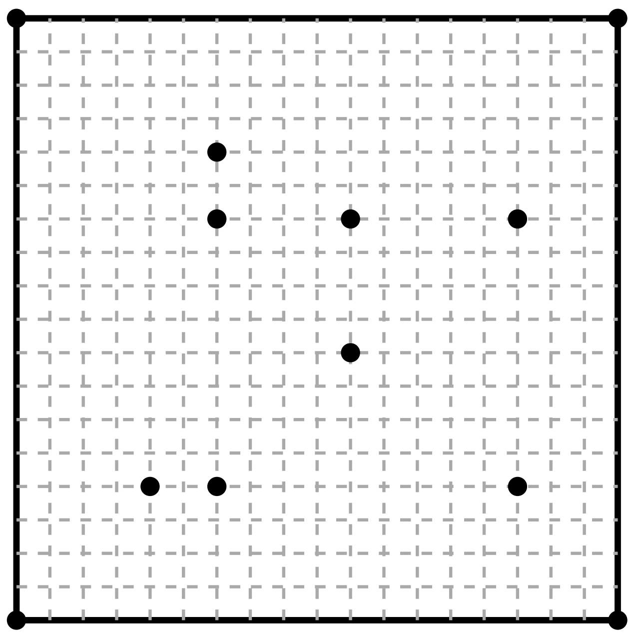




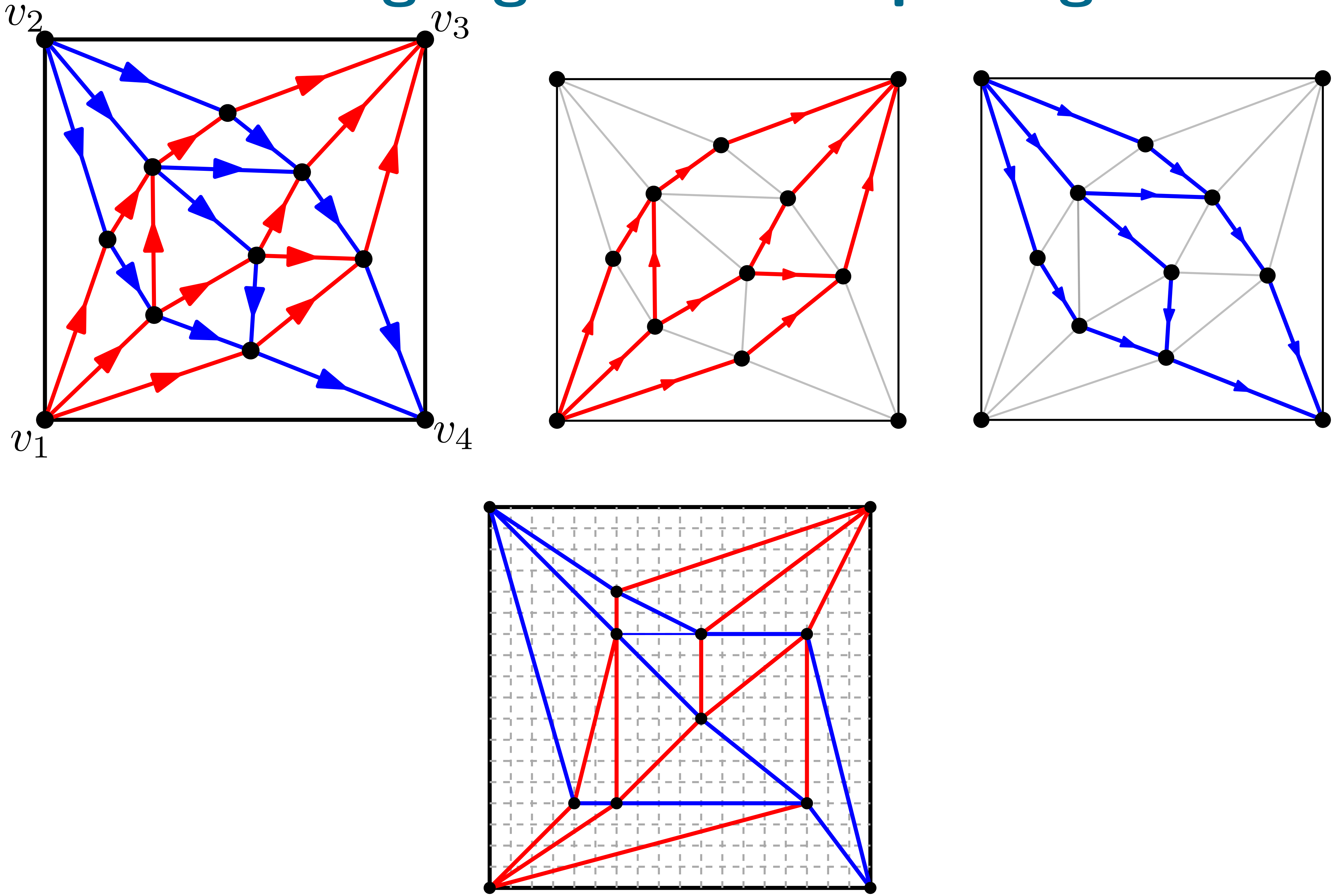


Face-counting algorithm on square grid

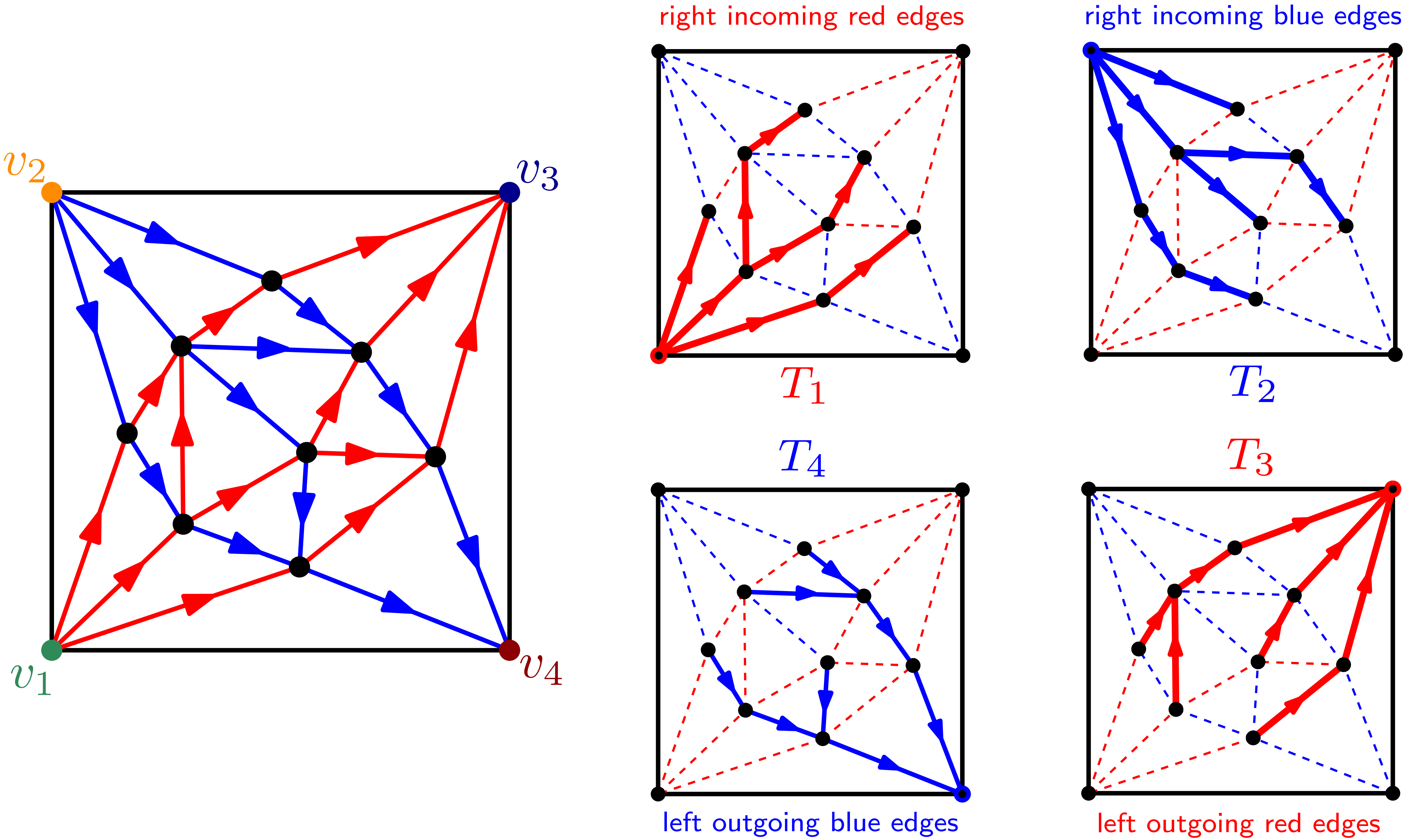




Face-counting algorithm on square grid

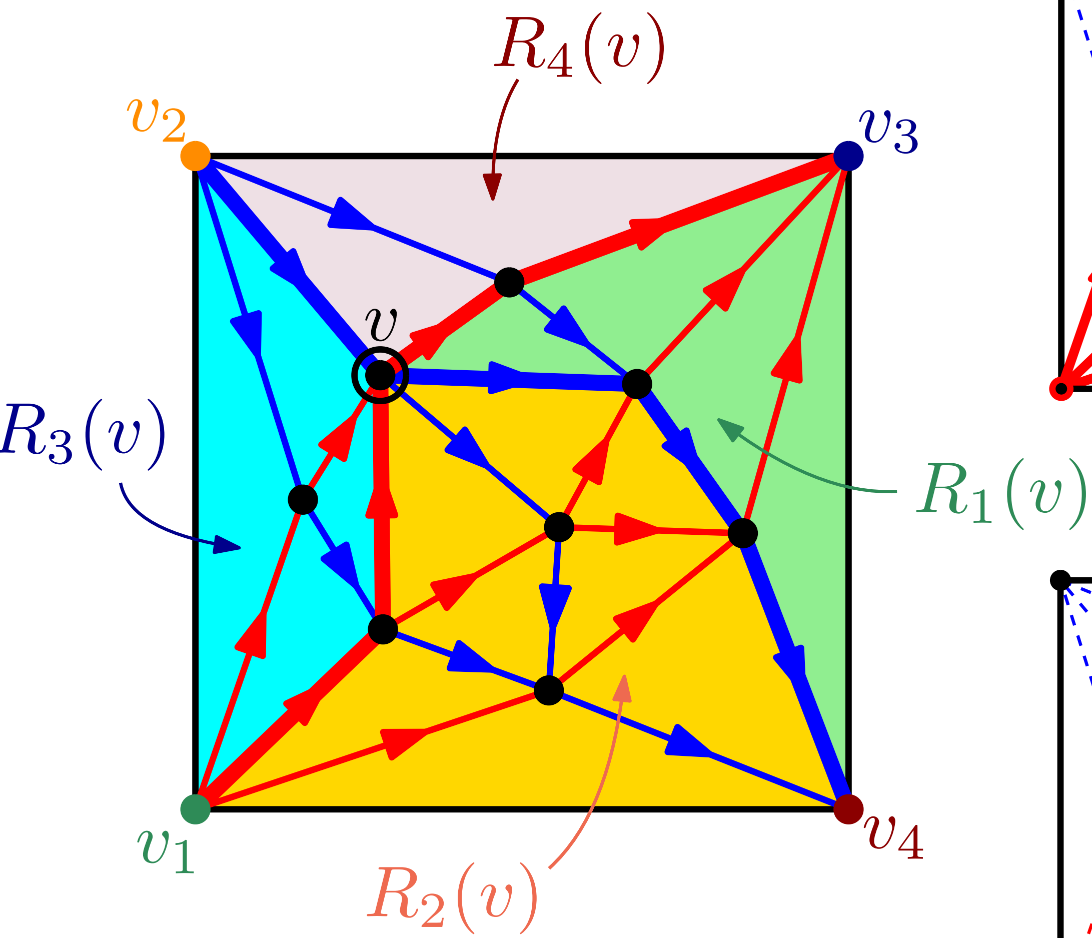


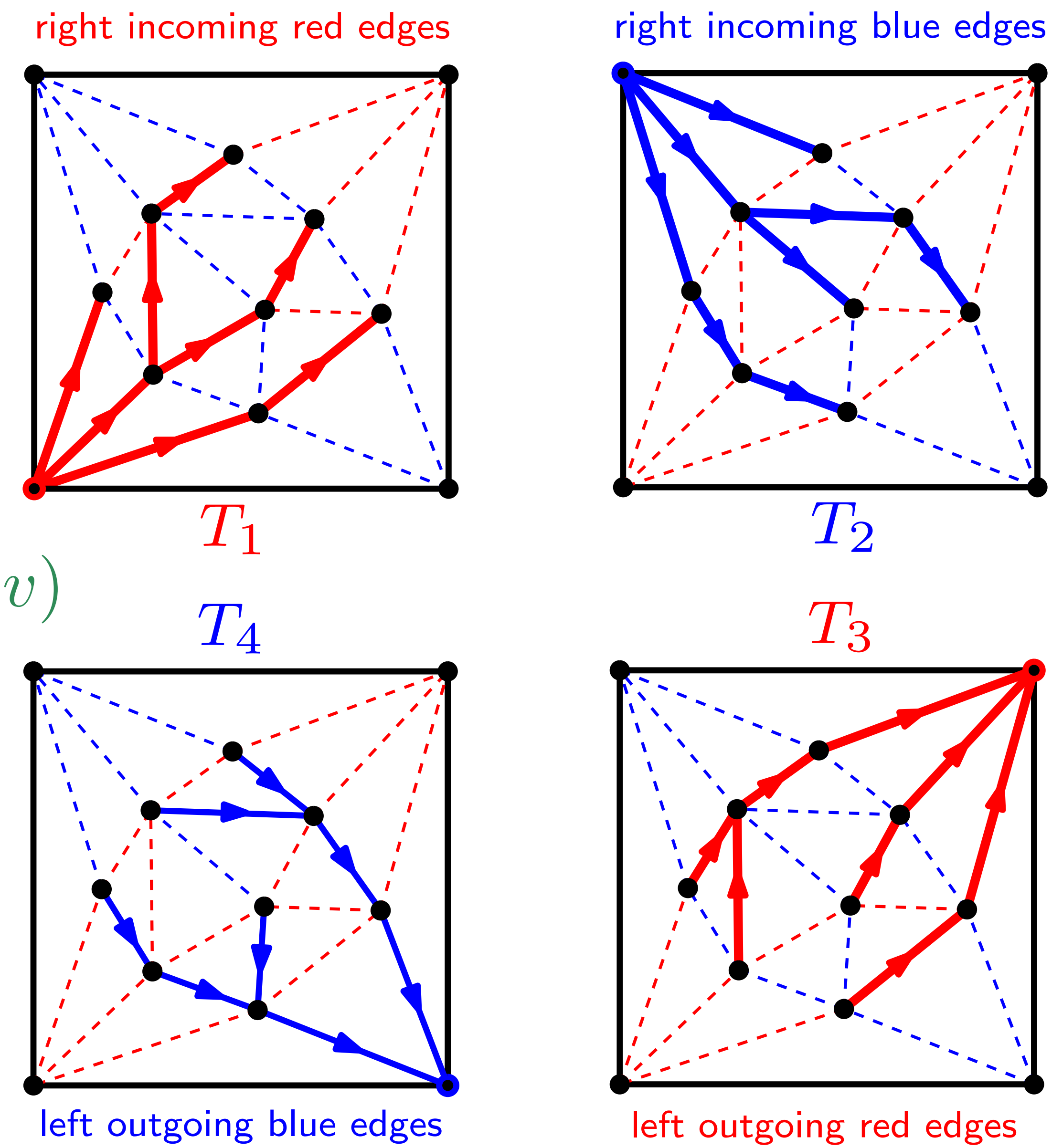
4-wood associated to transversal structure



4-wood associated to transversal structure

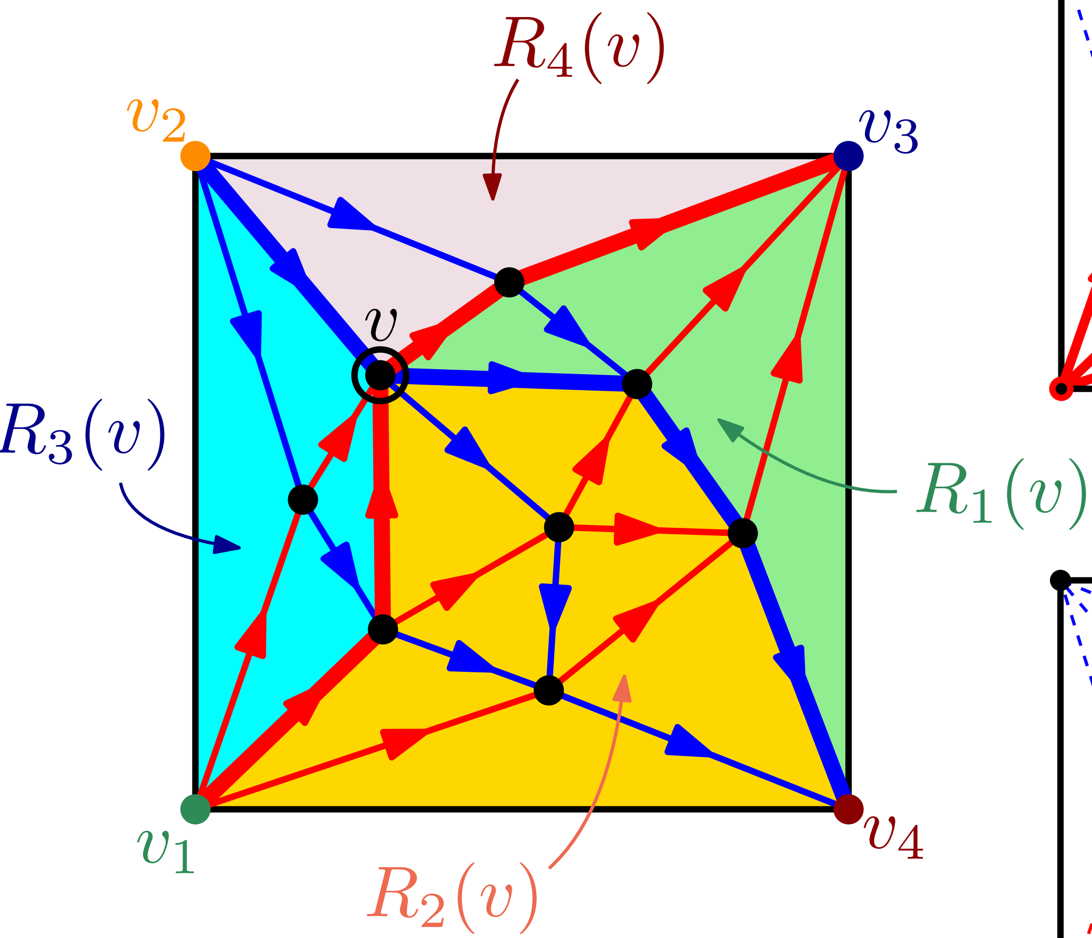
yields 4 regions
for each vertex v

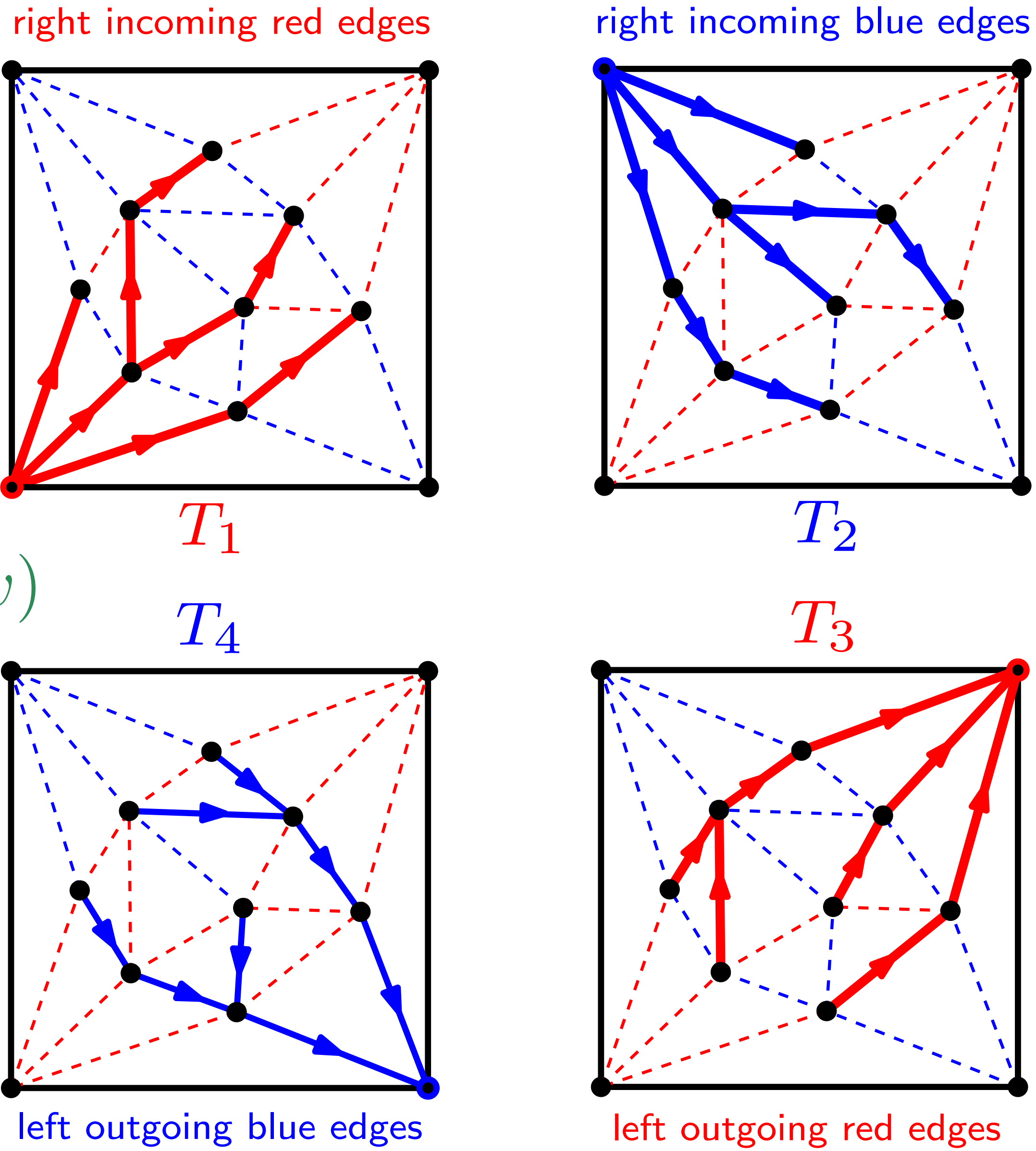




4-wood associated to transversal structure

yields 4 regions
for each vertex v





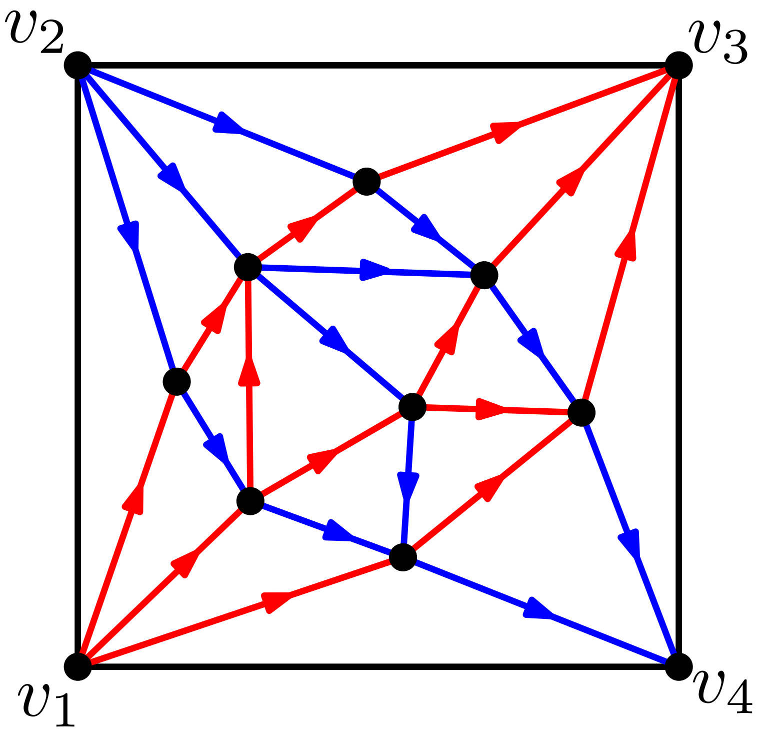
square-grid algo

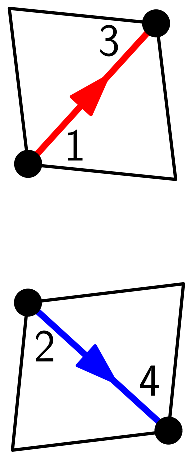
⇕

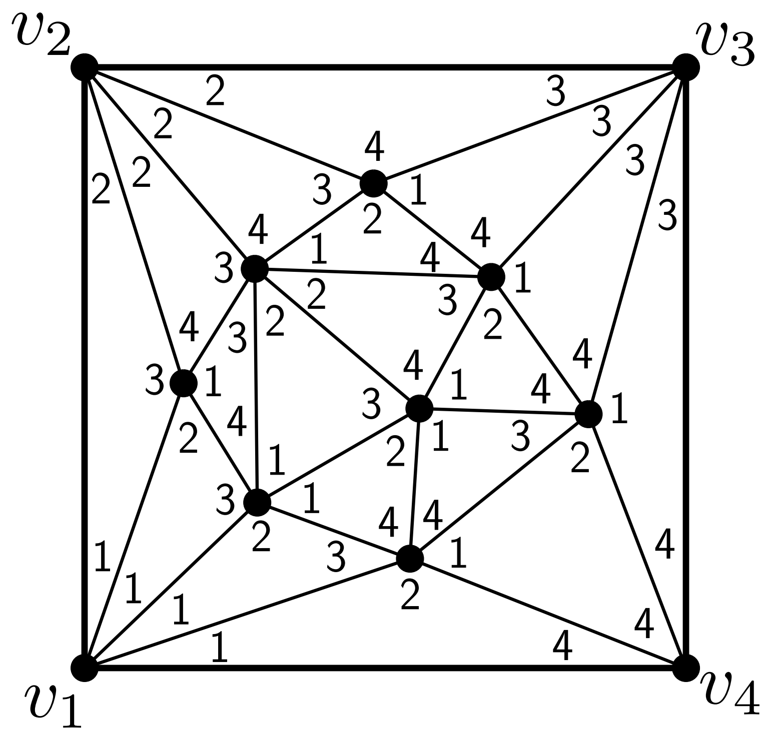
barycentric placement

(place v at $\frac{4}{28}v_1 + \frac{8}{28}v_2 + \frac{4}{28}v_3 + \frac{2}{28}v_4$)

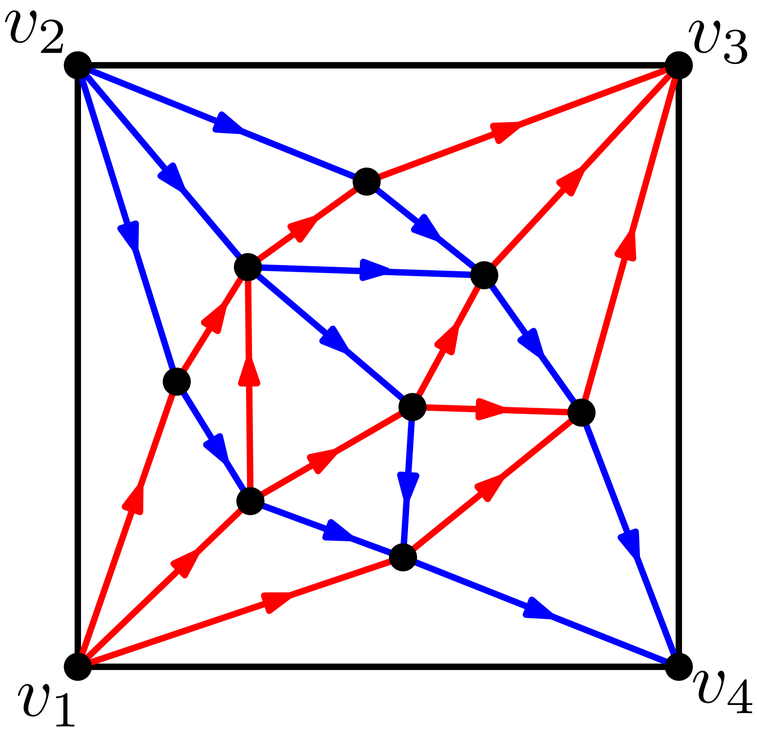
4-labeling associated to transversal structure

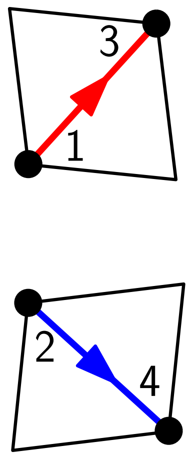


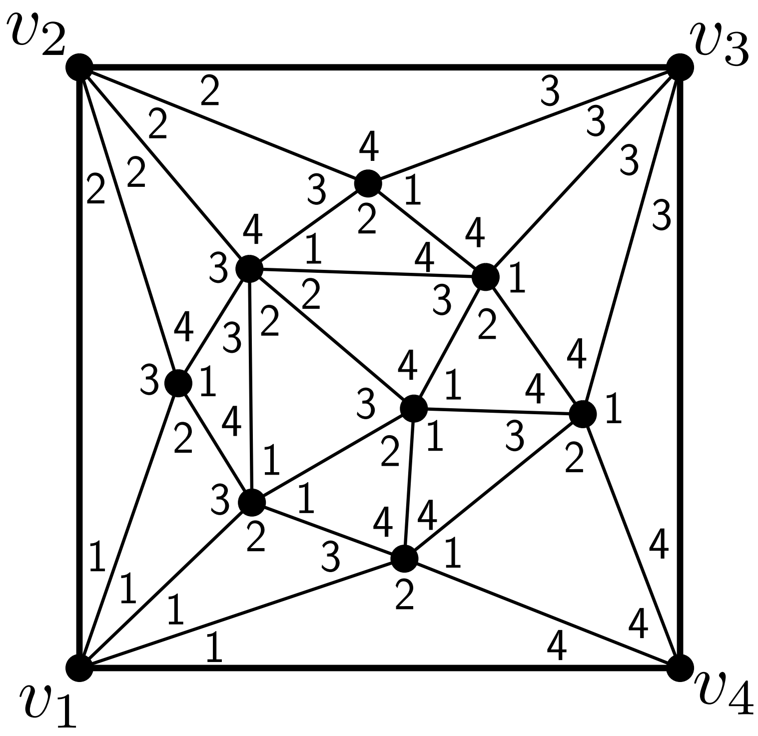


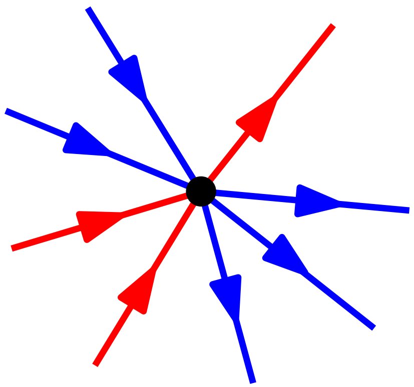


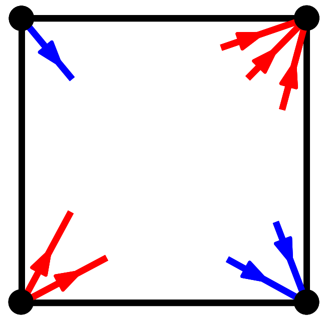
4-labeling associated to transversal structure

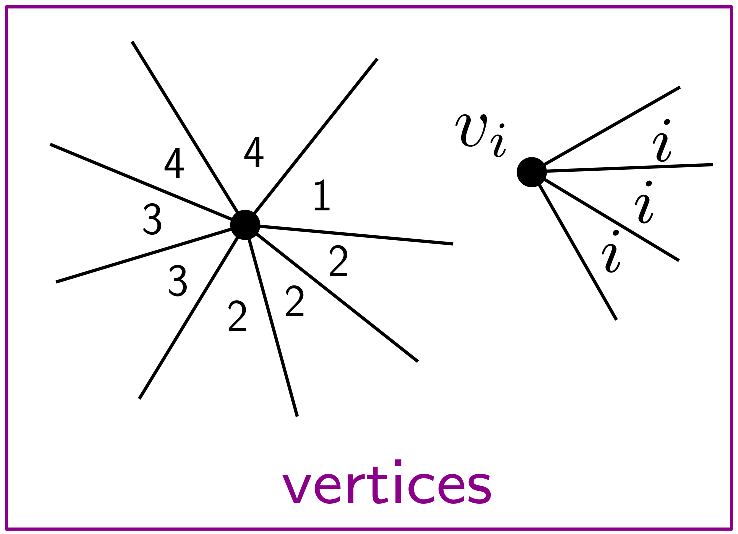




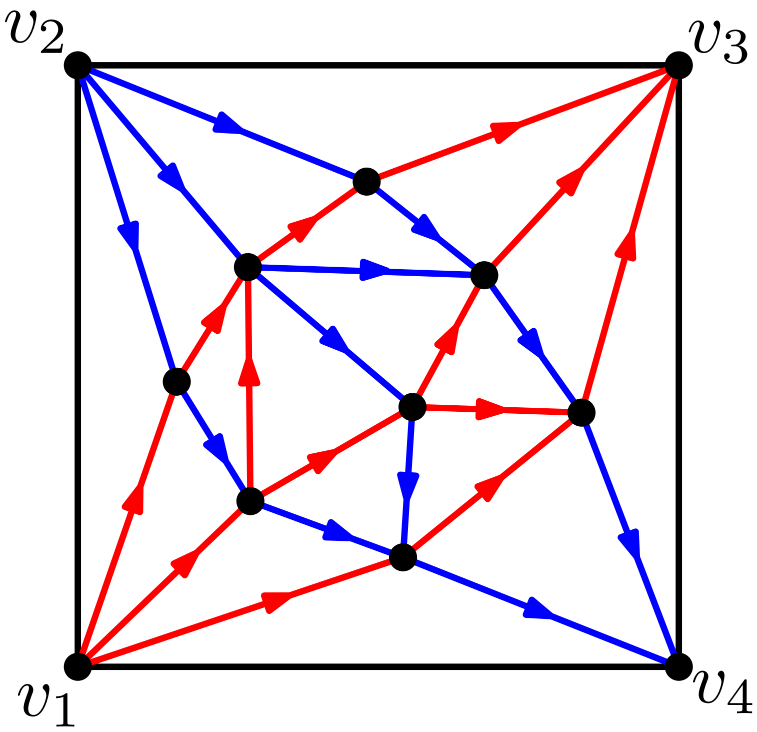


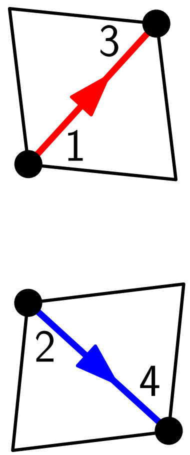


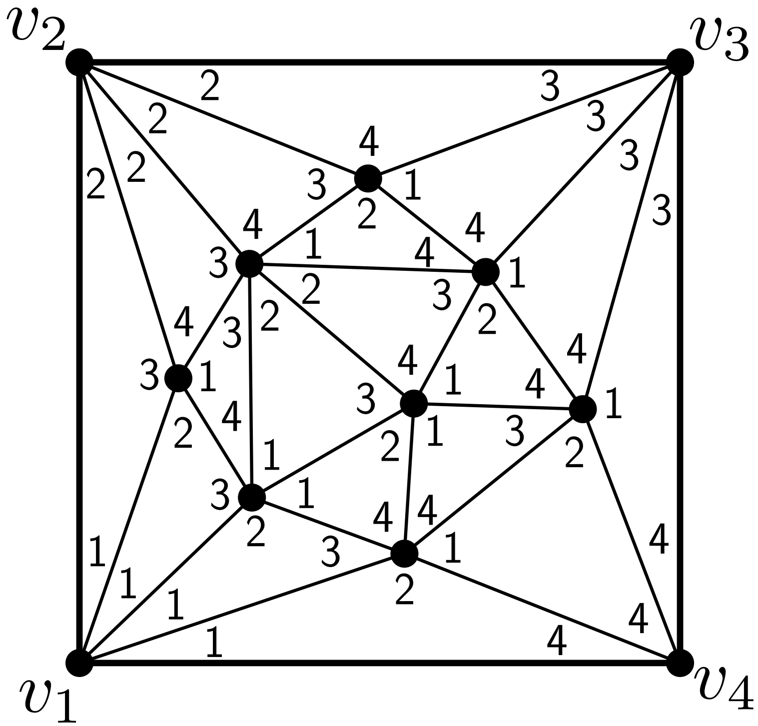


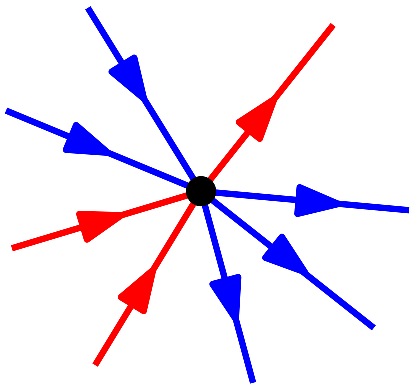


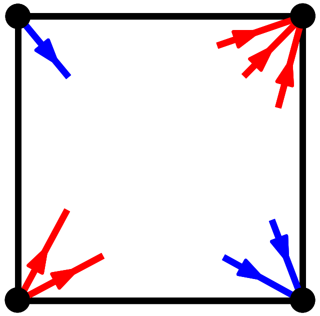
4-labeling associated to transversal structure

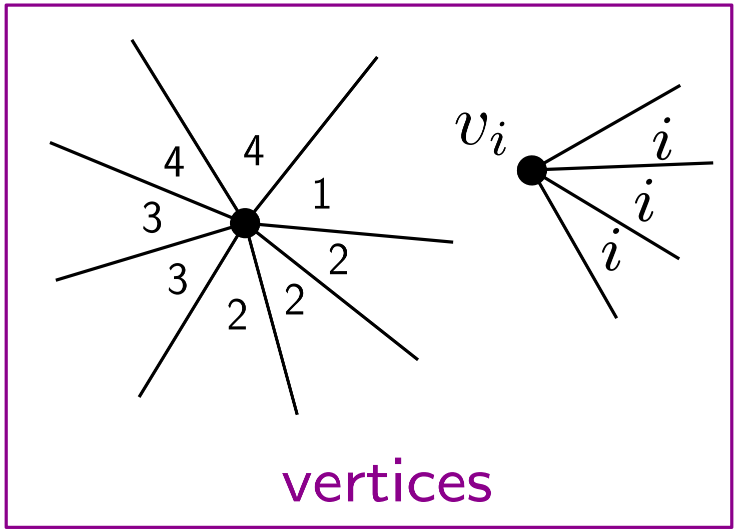


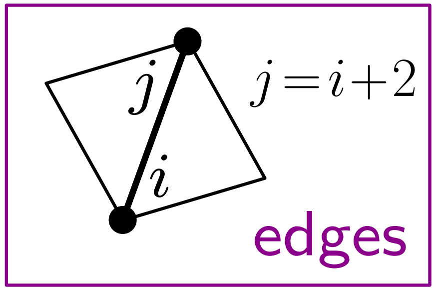




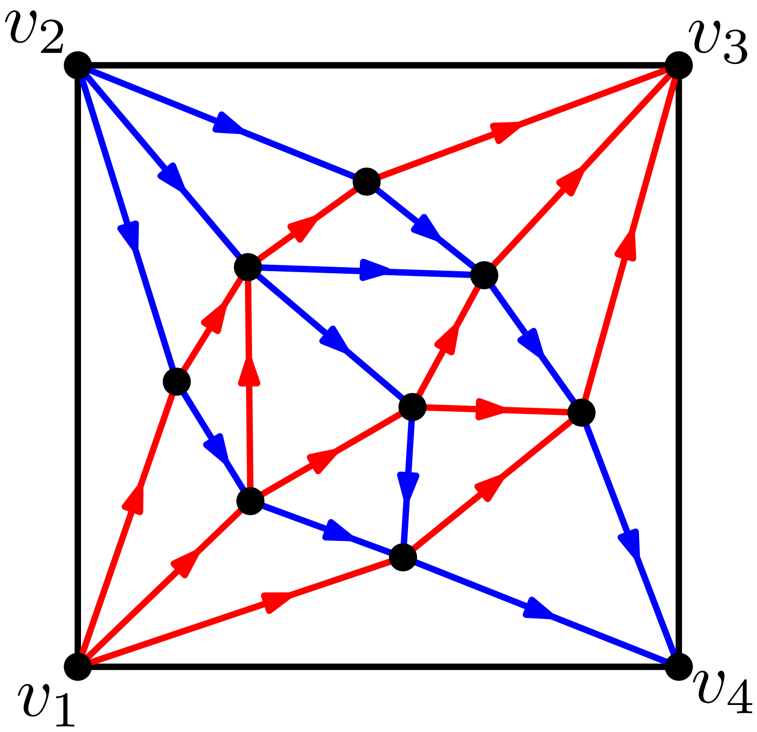


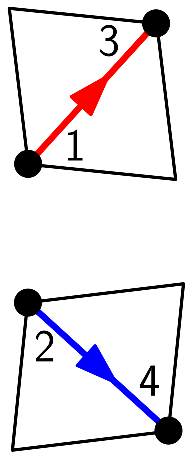


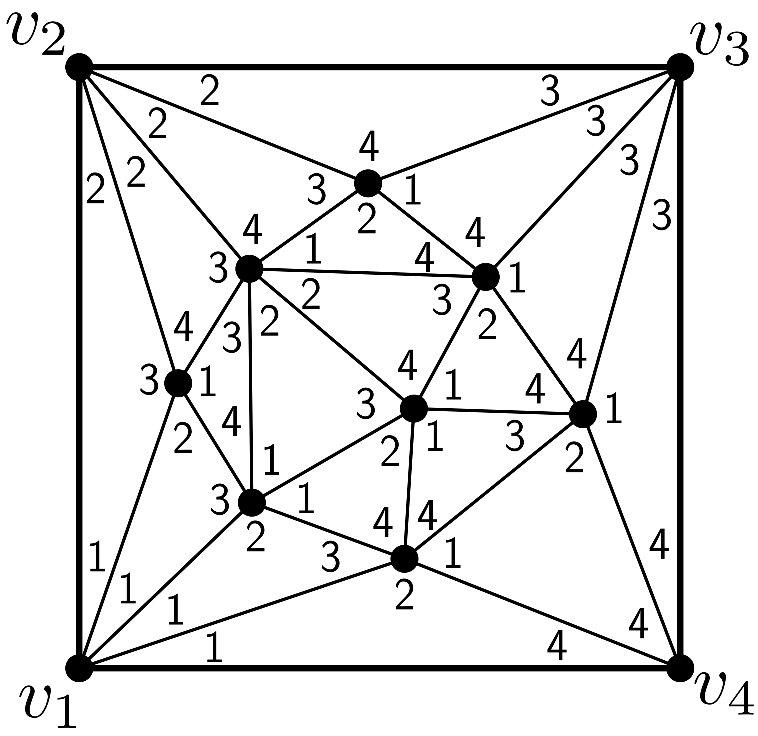


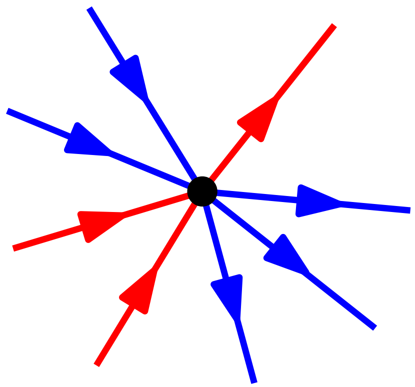


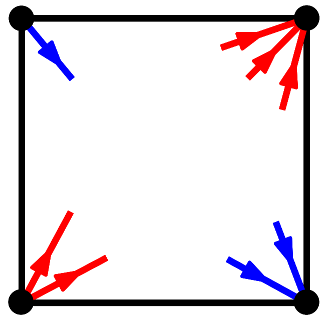
4-labeling associated to transversal structure

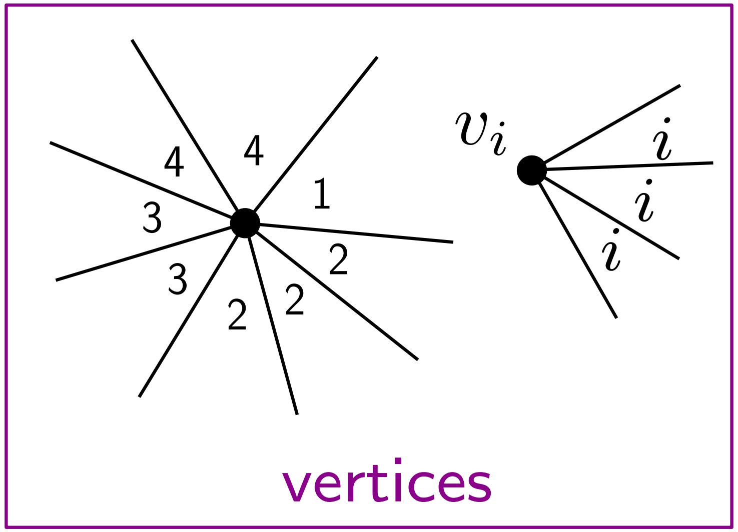


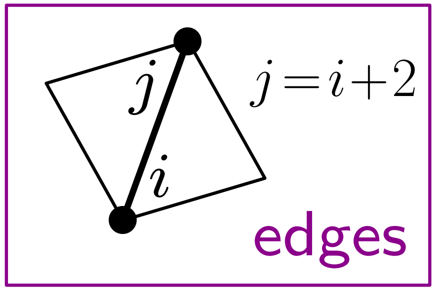


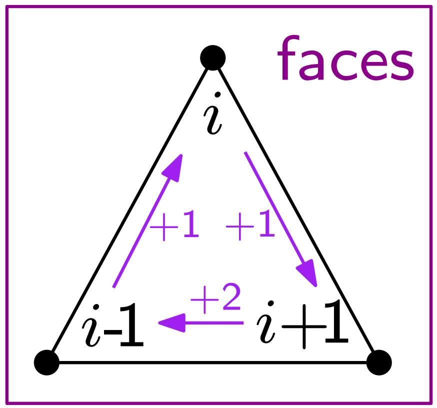




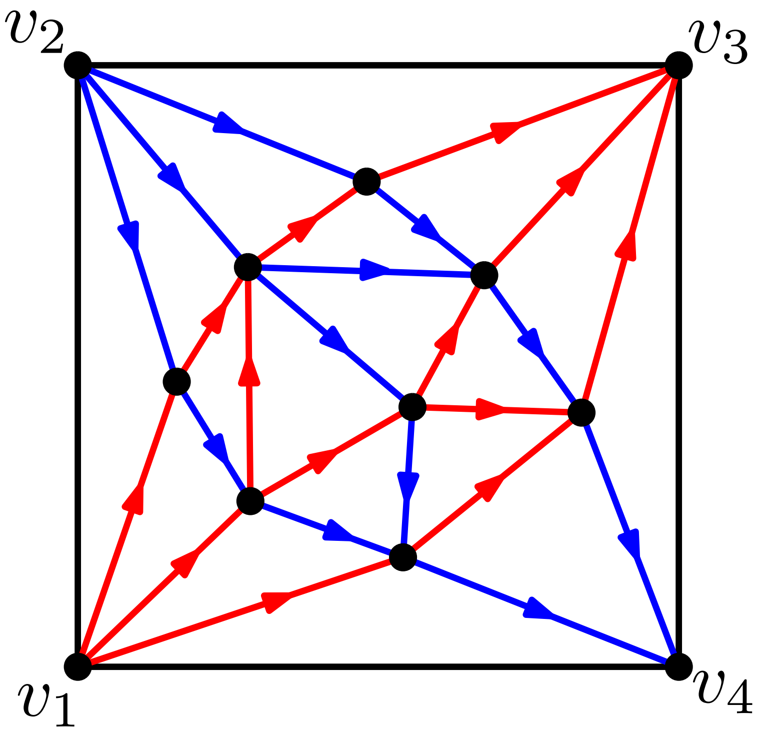


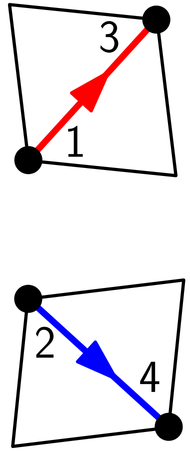


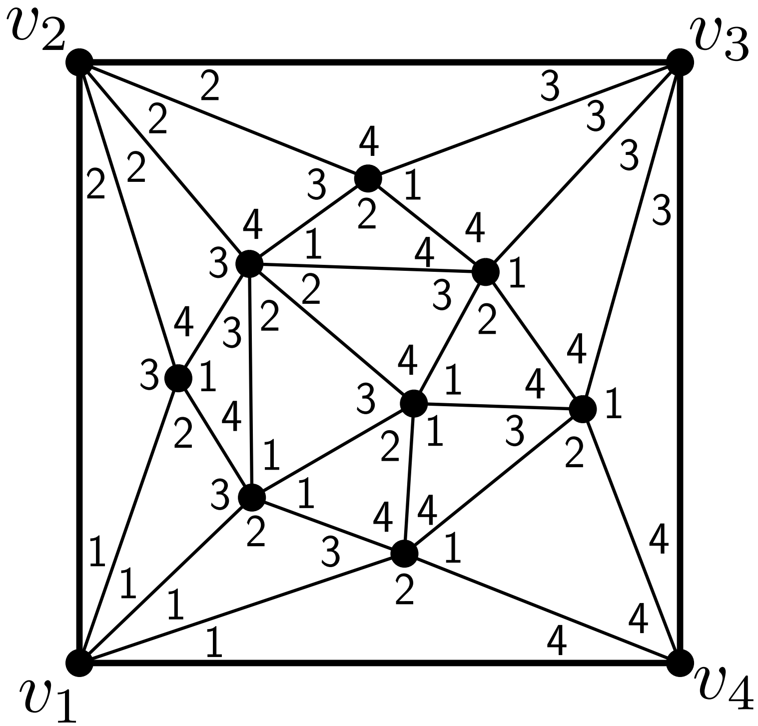


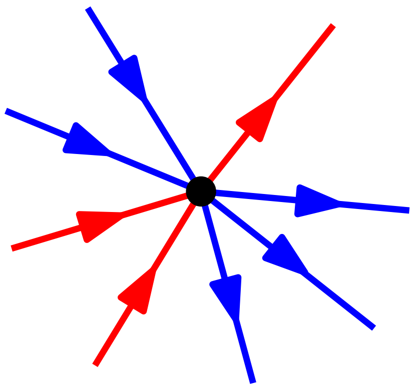


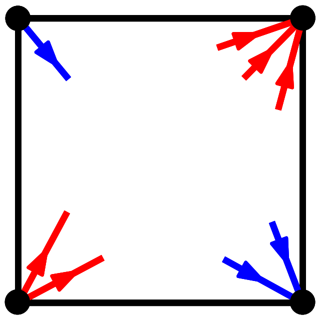
4-labeling associated to transversal structure

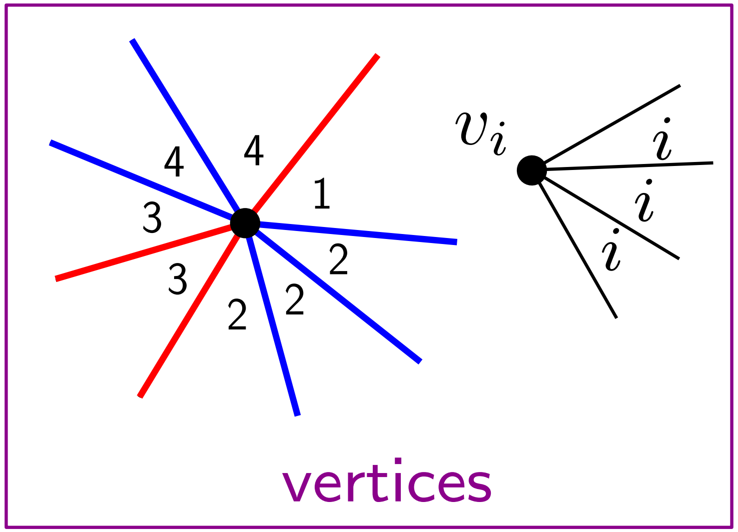


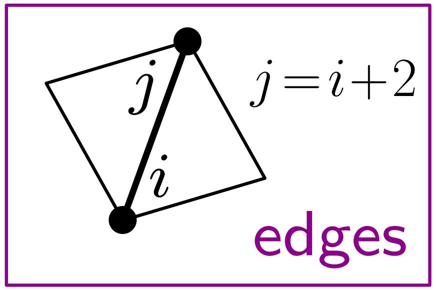


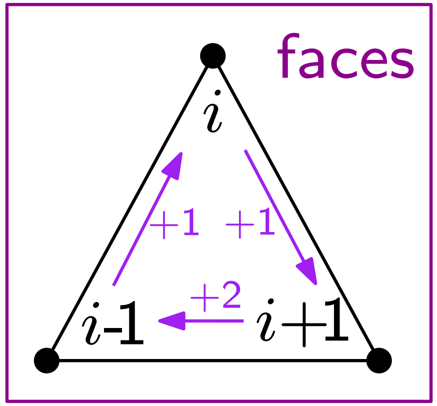




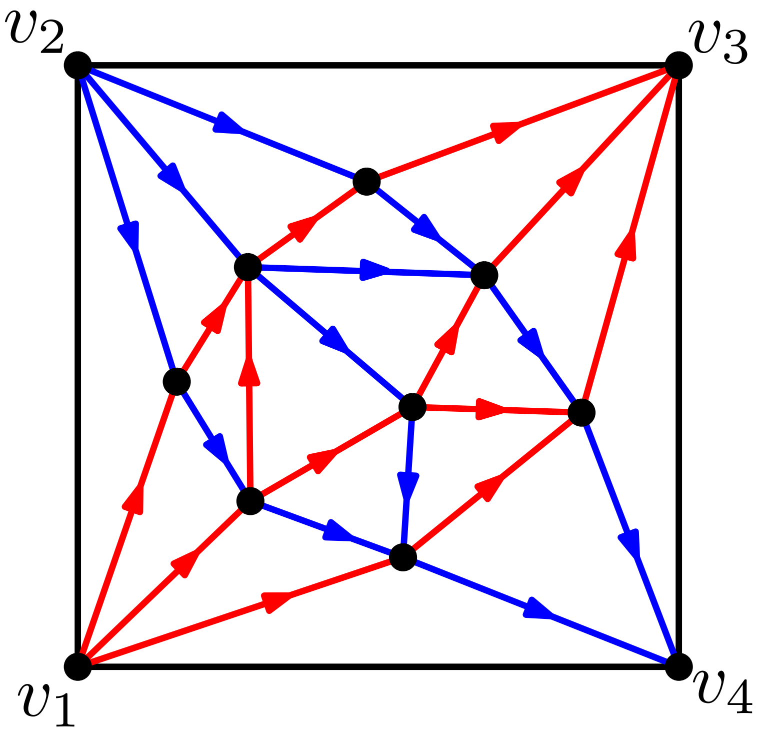


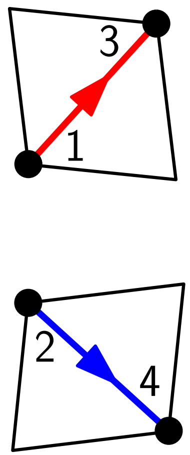


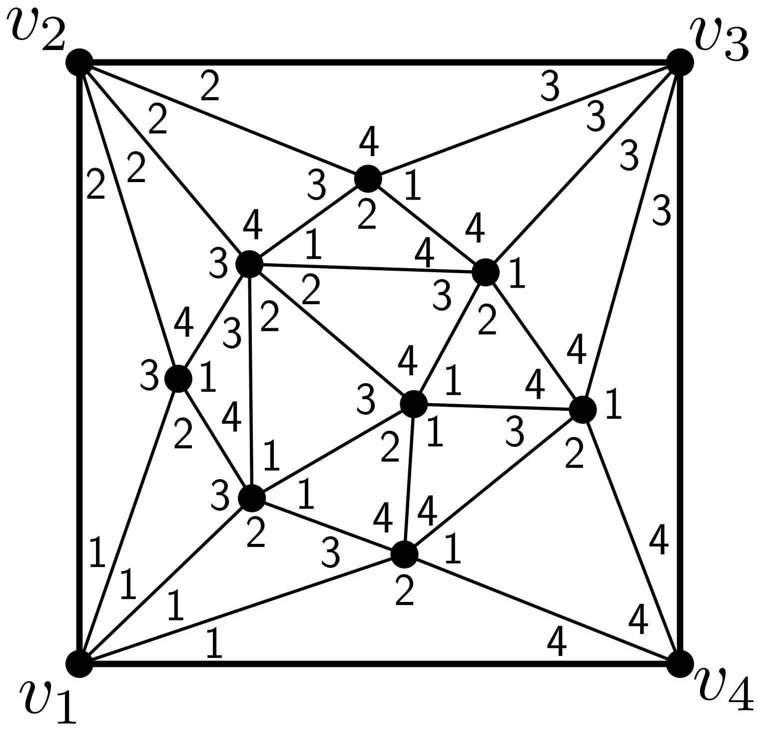


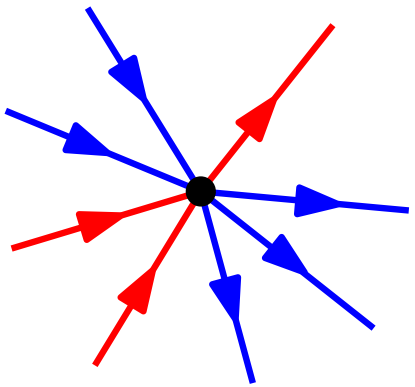


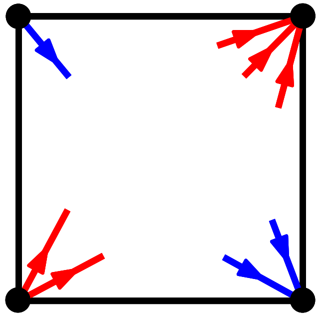
4-labeling associated to transversal structure

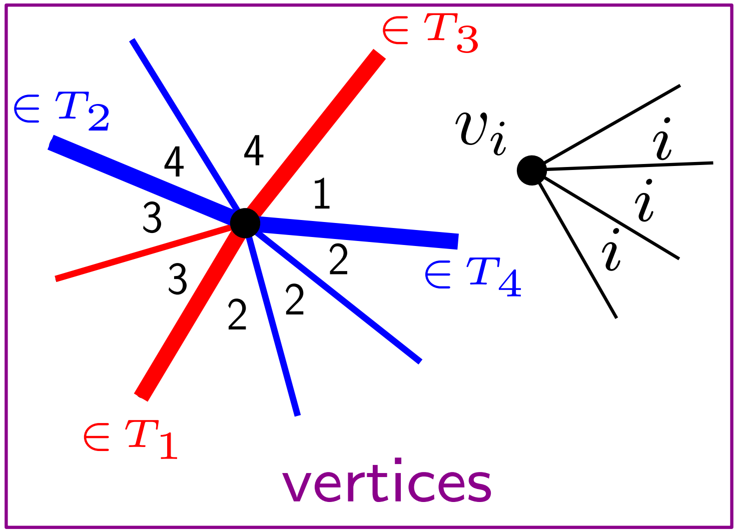


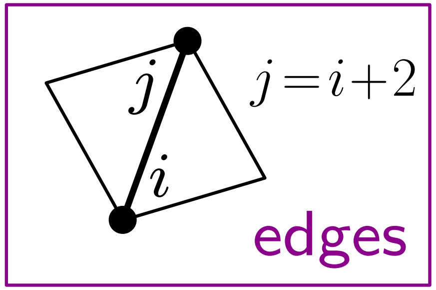


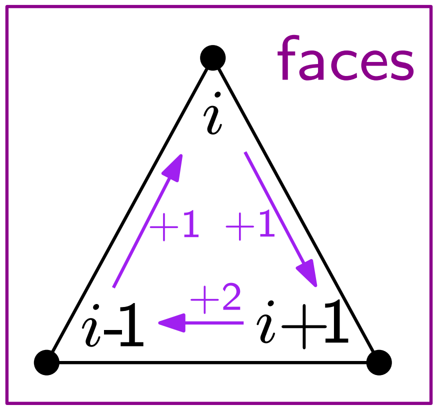








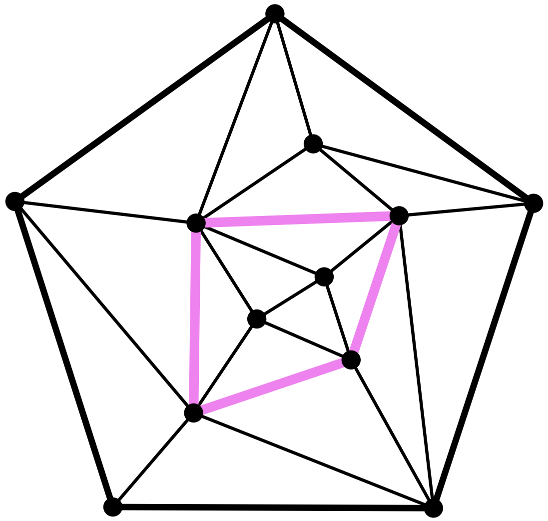




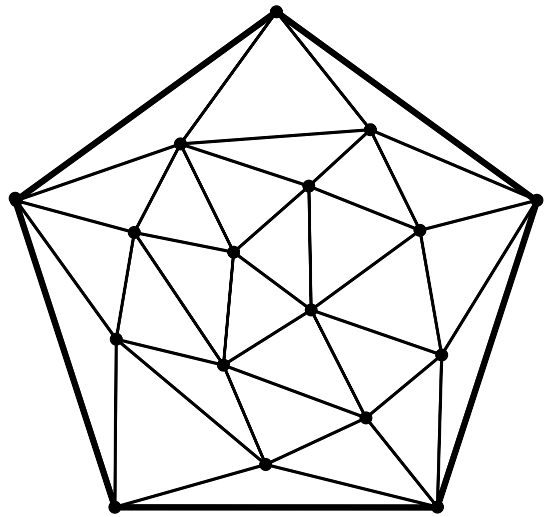
5-connected case

5c-triangulations

5c-triangulation = triangulation of 5-gon such that every cycle with at least one vertex inside has length ≥ 5



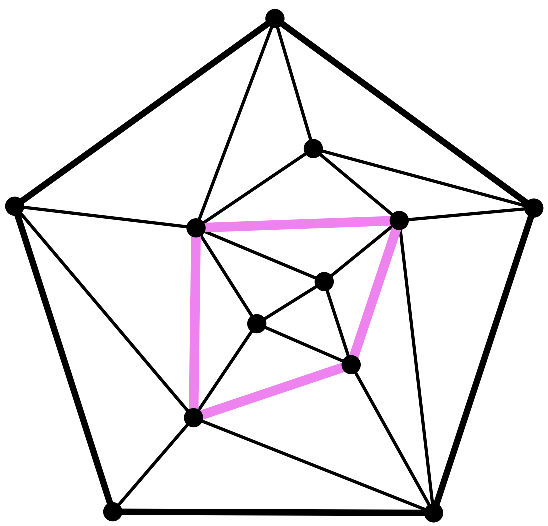
not 5c-triangulation



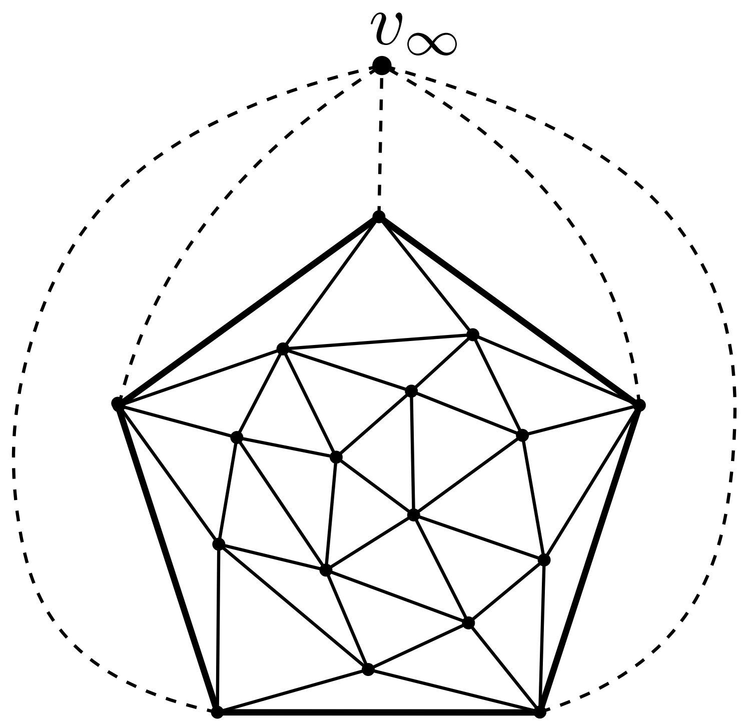
5c-triangulation

5c-triangulations

5c-triangulation = triangulation of 5-gon such that every cycle with at least one vertex inside has length ≥ 5



not 5c-triangulation



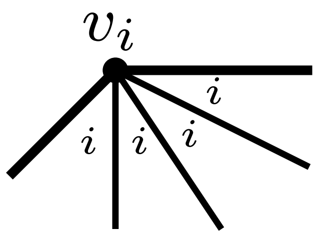
5c-triangulation

$\Updownarrow \approx$

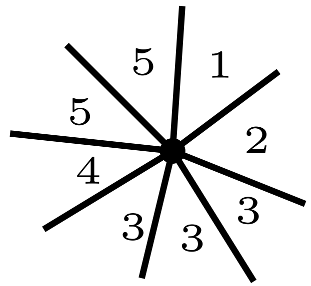
triangulation augmented by v_∞ is 5-connected

5c-labelings

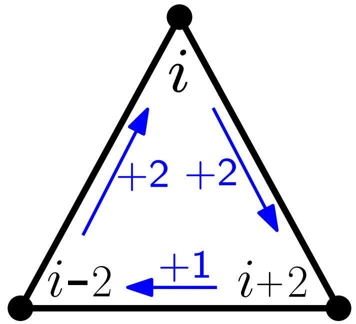
Any 5c-triangulation has a **labeling** of corners by $\{1, 2, 3, 4, 5\}$ so that



outer vertices

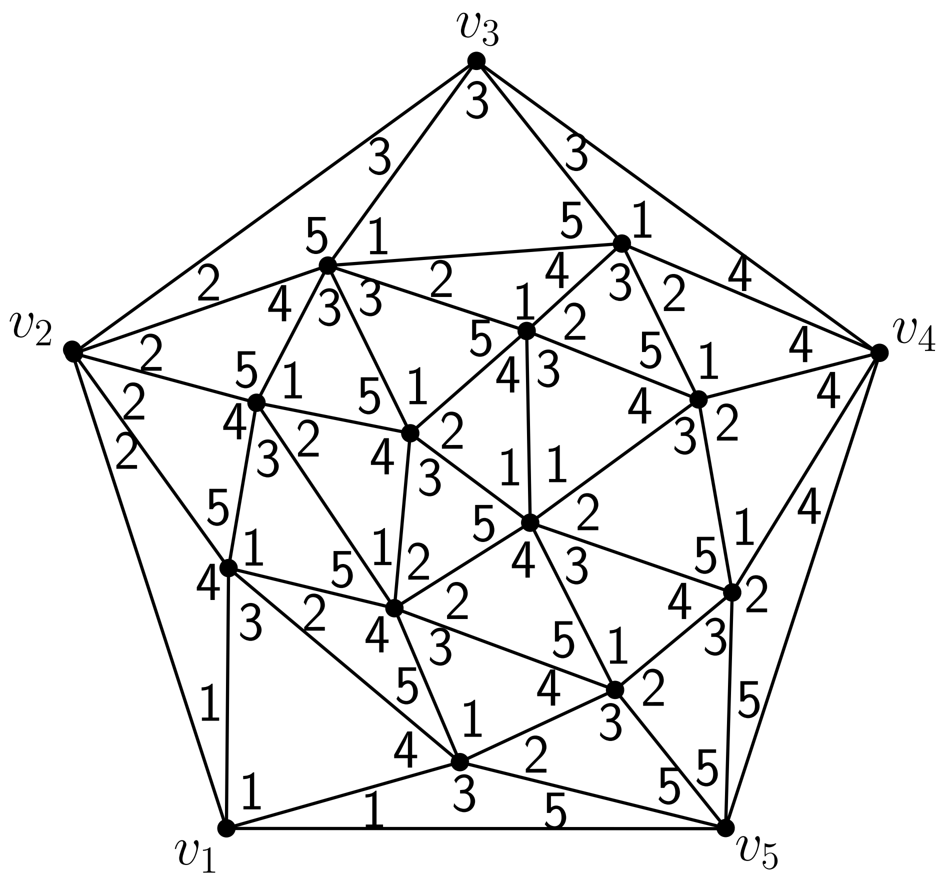


inner vertices



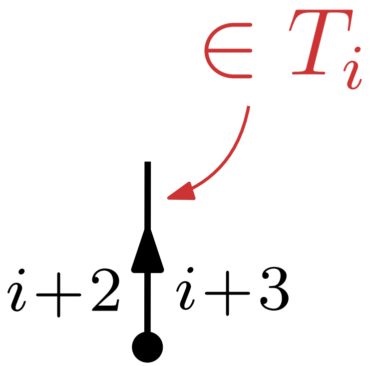
inner faces

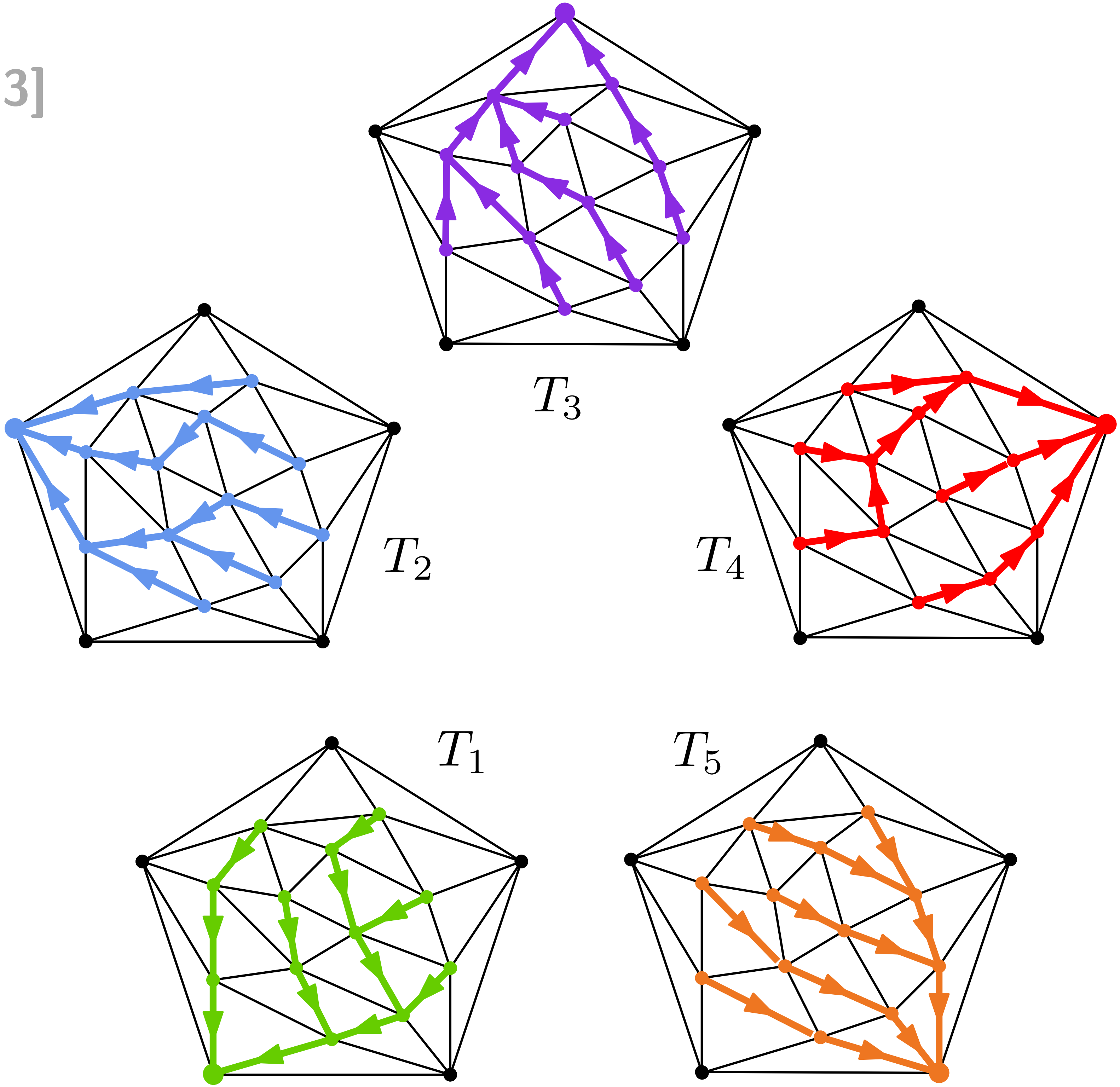
5c-labeling



5c-woods

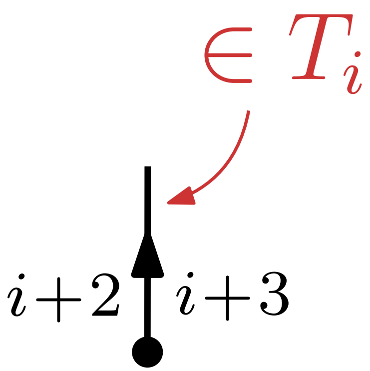
[Bernardi,F,Liang'23]

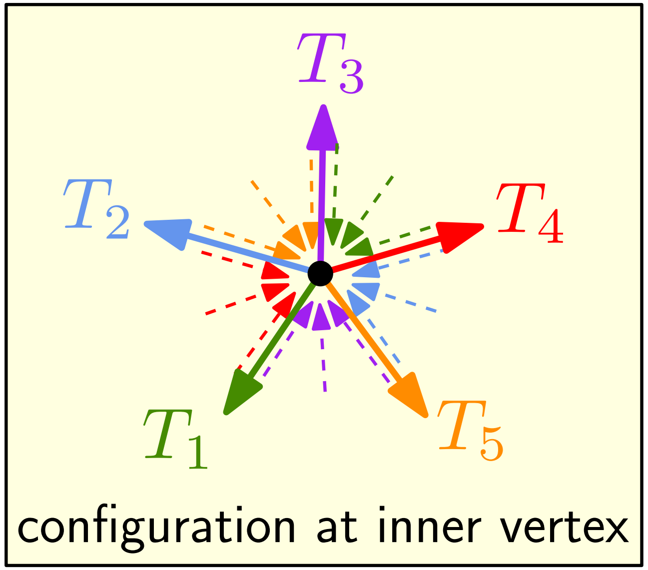


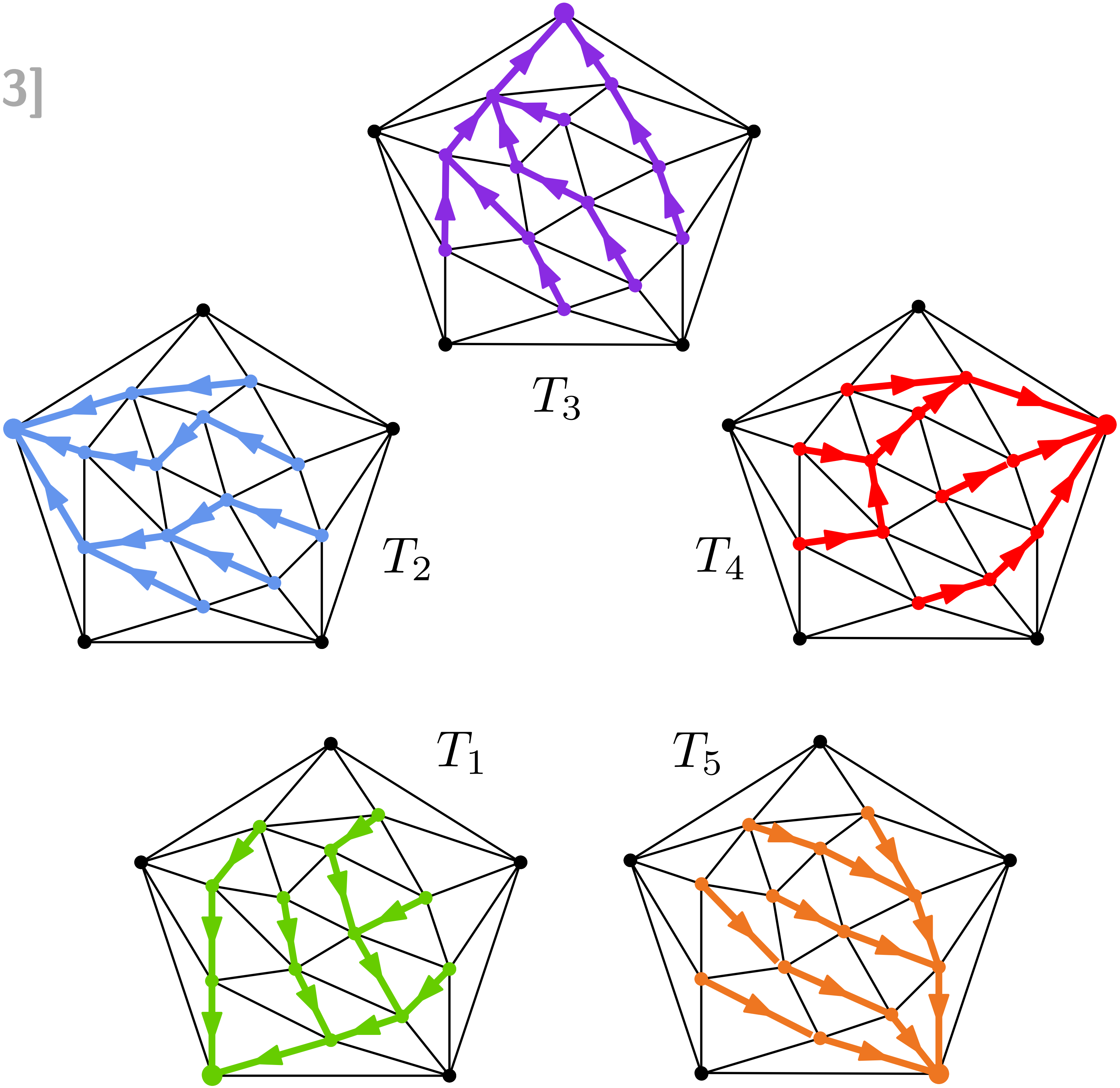


5c-woods

[Bernardi,F,Liang'23]

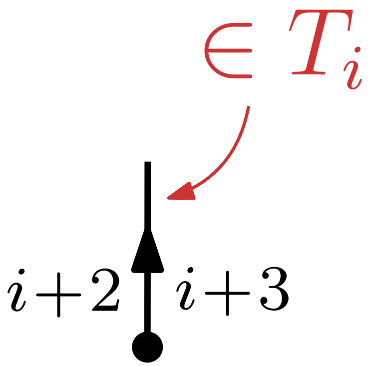


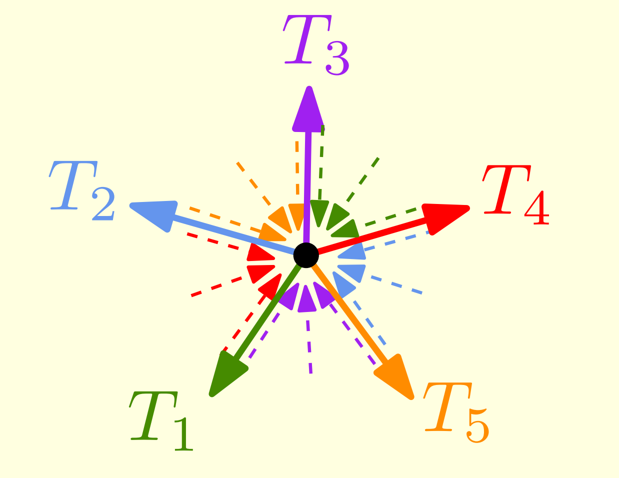




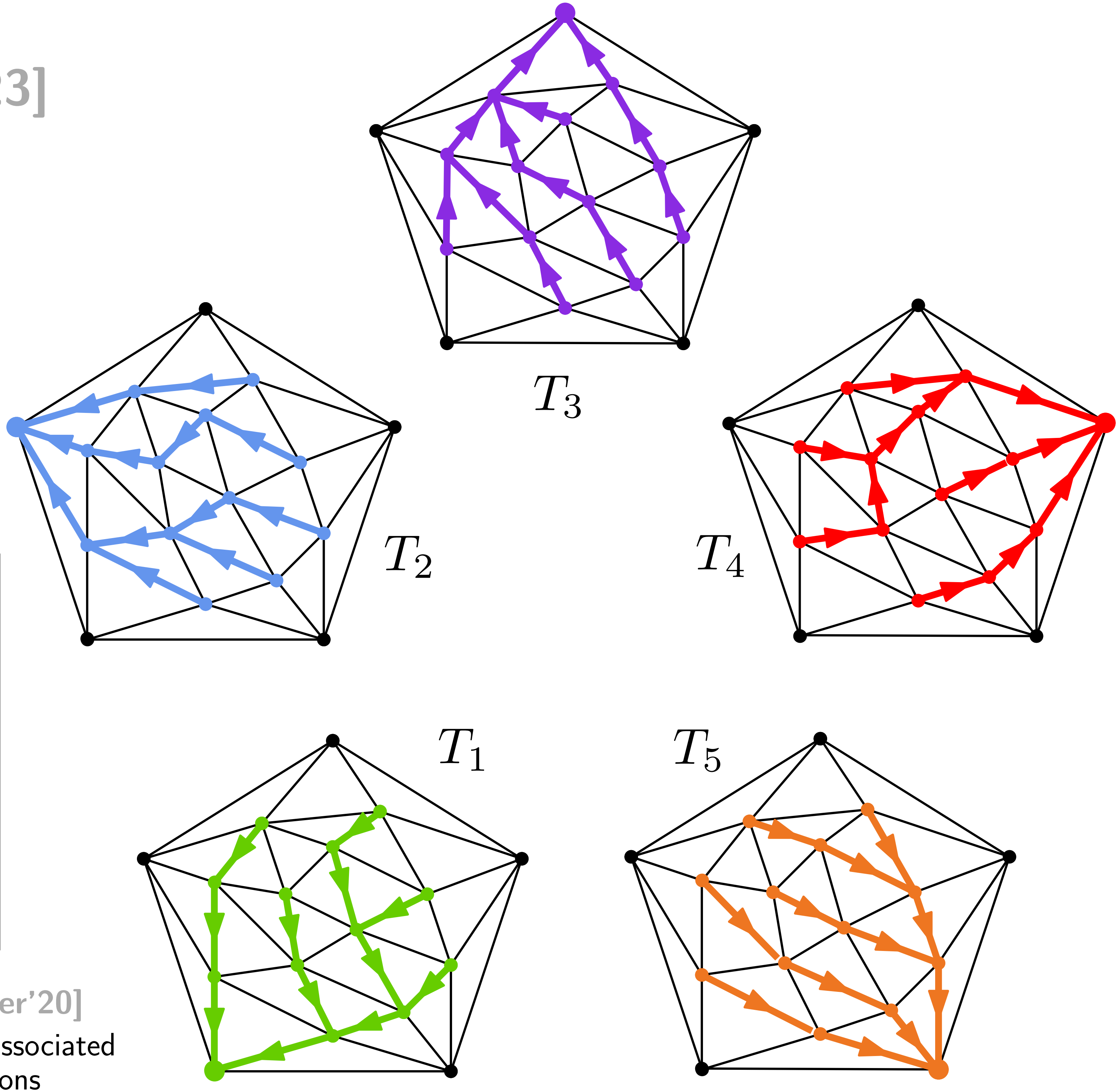
5c-woods

[Bernardi,F,Liang'23]





configuration at inner vertex

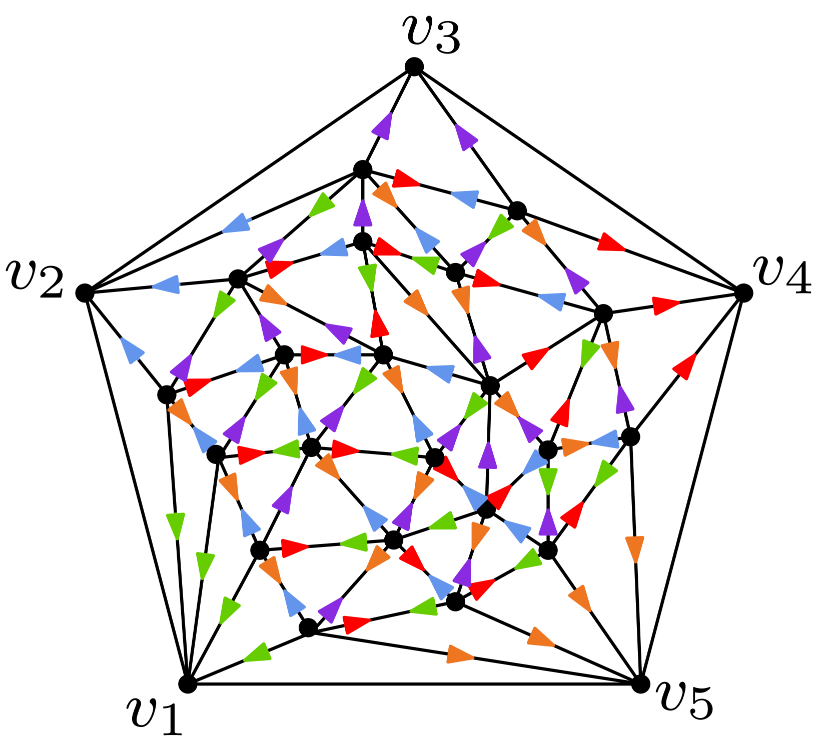


[Felsner, Schrezenmaier, Steiner'20]

other 5-woods (less restrictive) associated to pentagon-contact representations

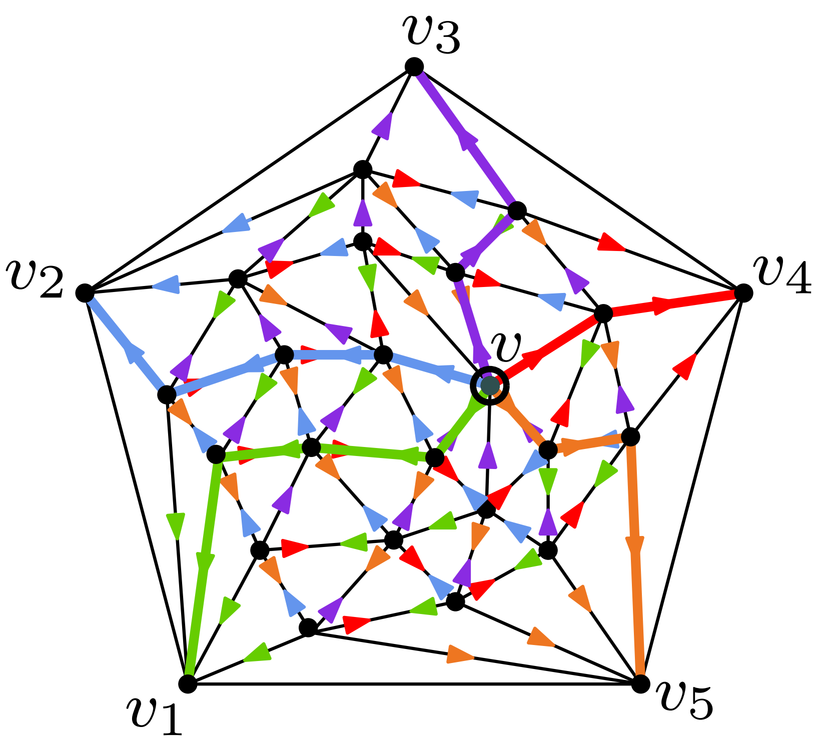
Face-counting algorithm

[Bernardi,F,Liang'23]



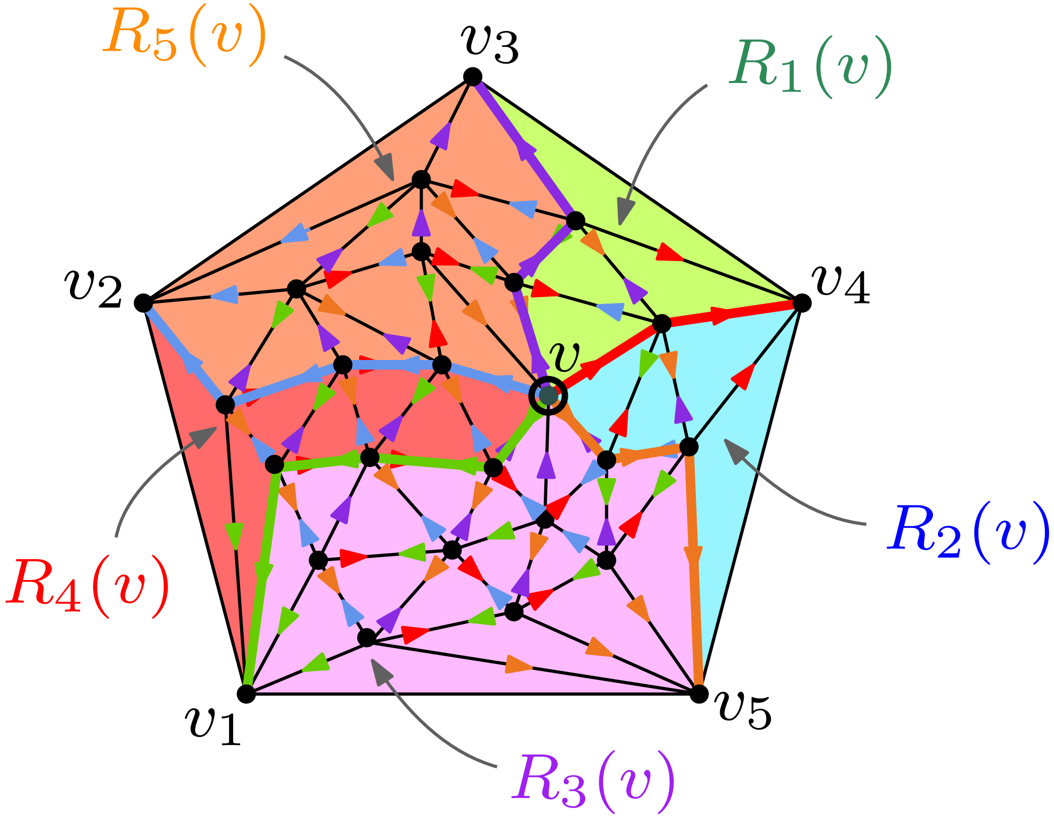
Face-counting algorithm

[Bernardi,F,Liang'23]



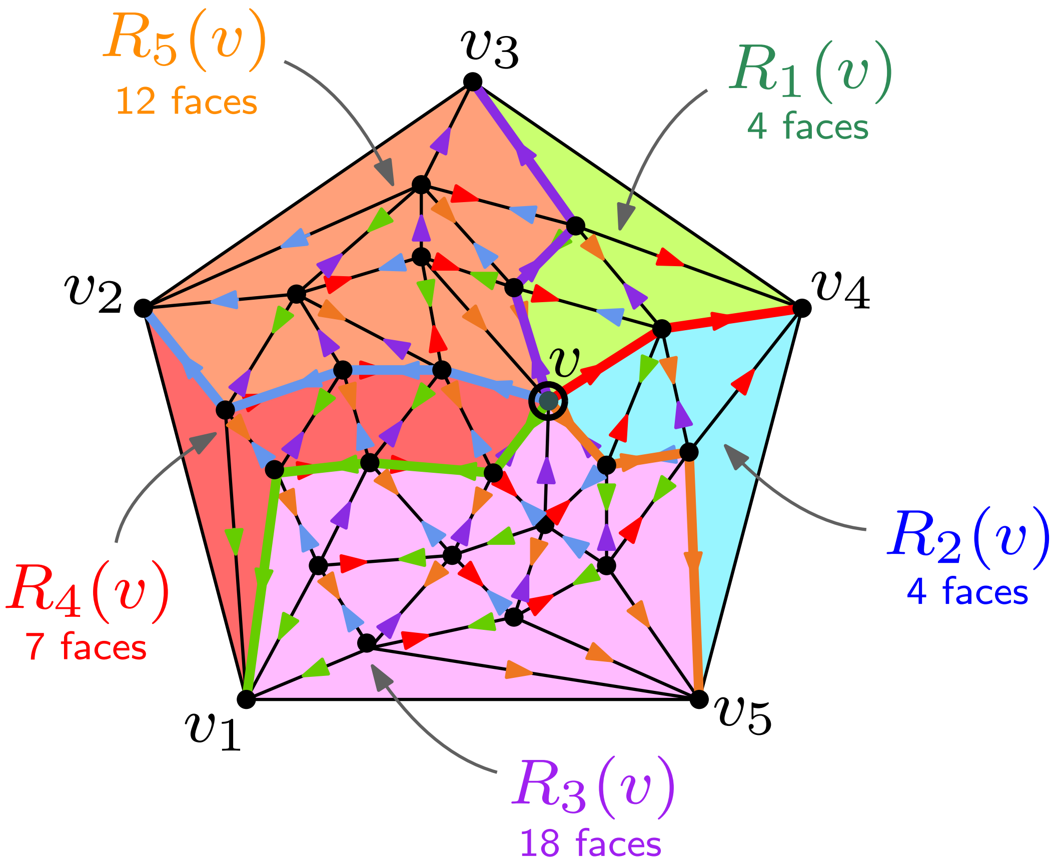
Face-counting algorithm

[Bernardi,F,Liang'23]

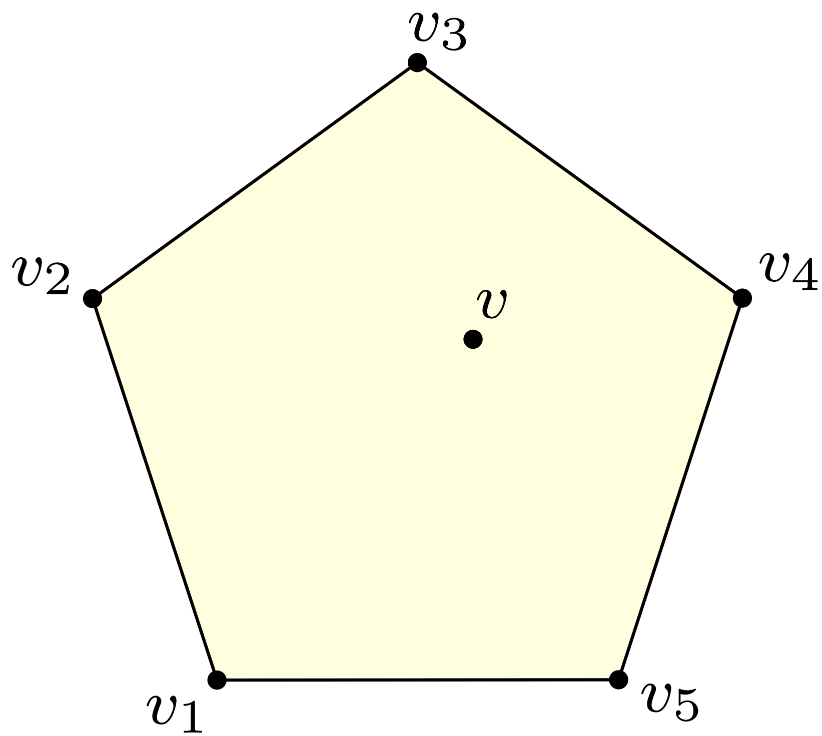


Face-counting algorithm

[Bernardi,F,Liang'23]



45 inner faces in total

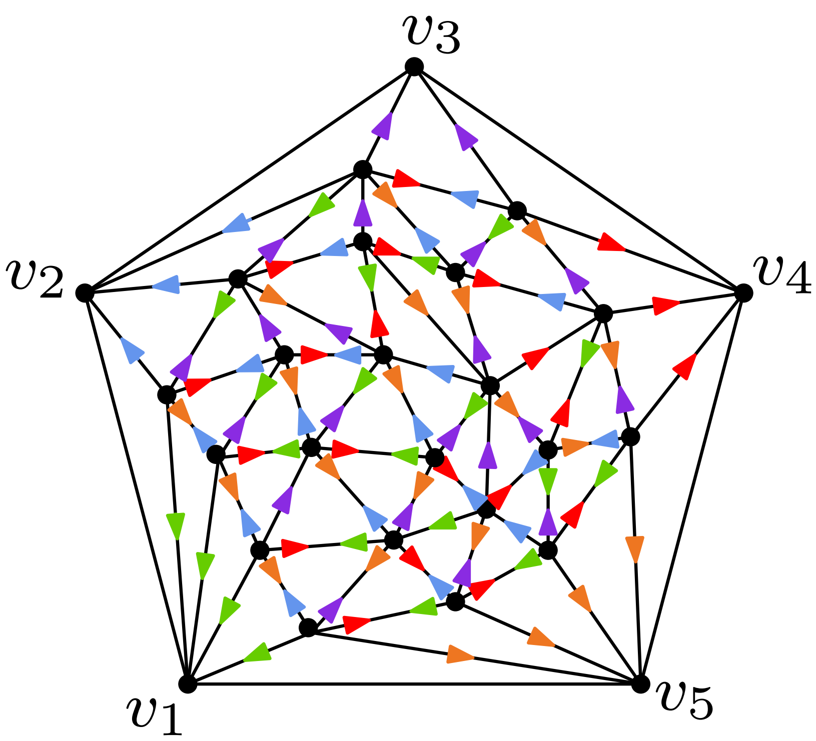


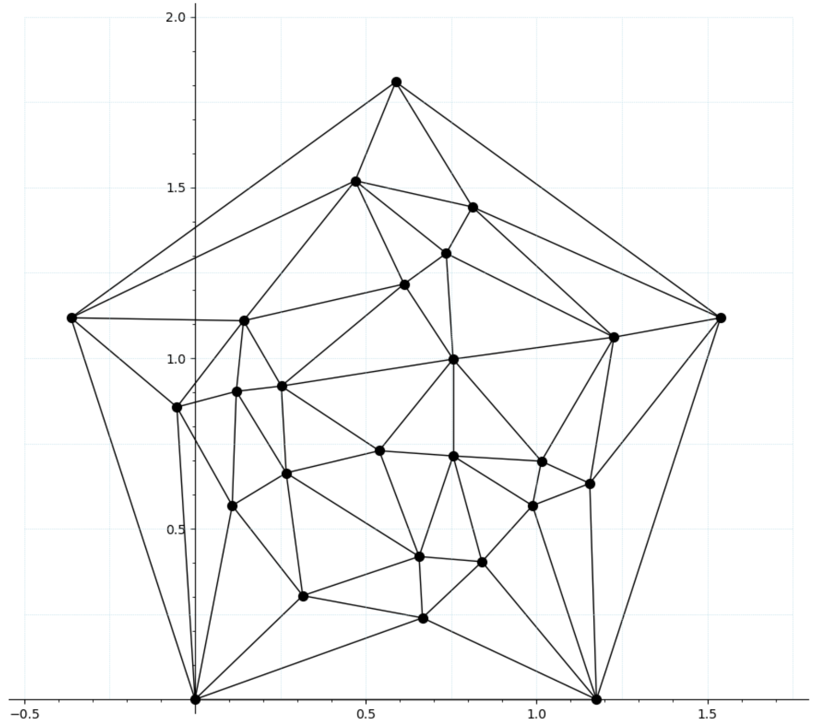
place v at

$$\frac{4}{45}v_1 + \frac{4}{45}v_2 + \frac{18}{45}v_3 + \frac{7}{45}v_4 + \frac{12}{45}v_5$$

Face-counting algorithm

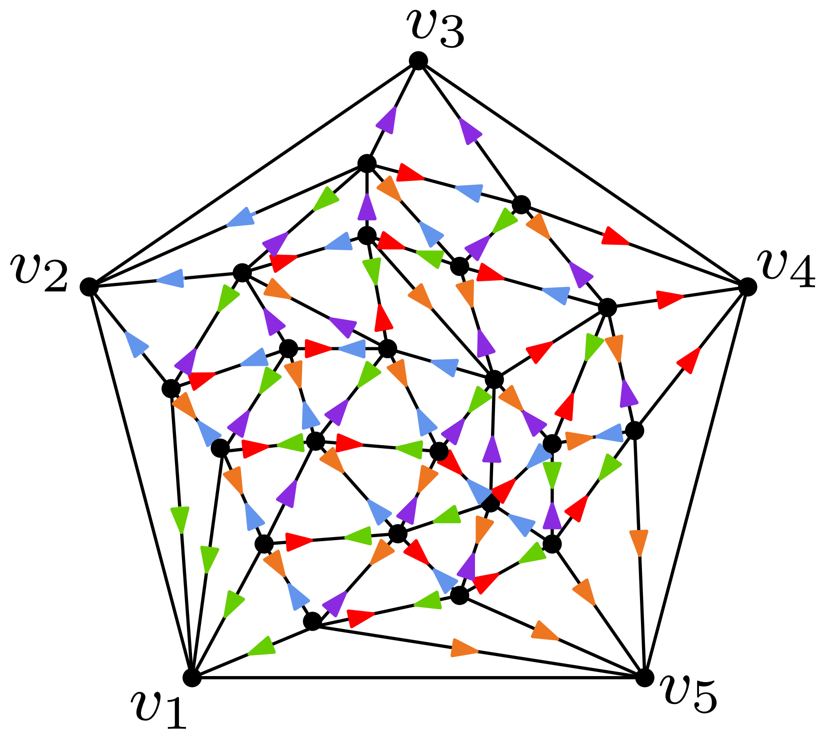
[Bernardi,F,Liang'23]

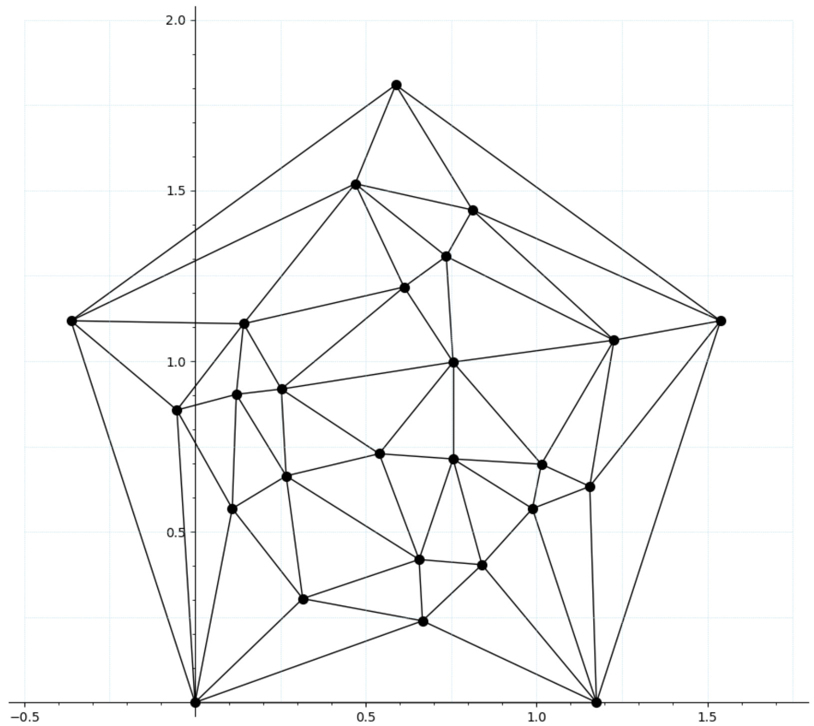




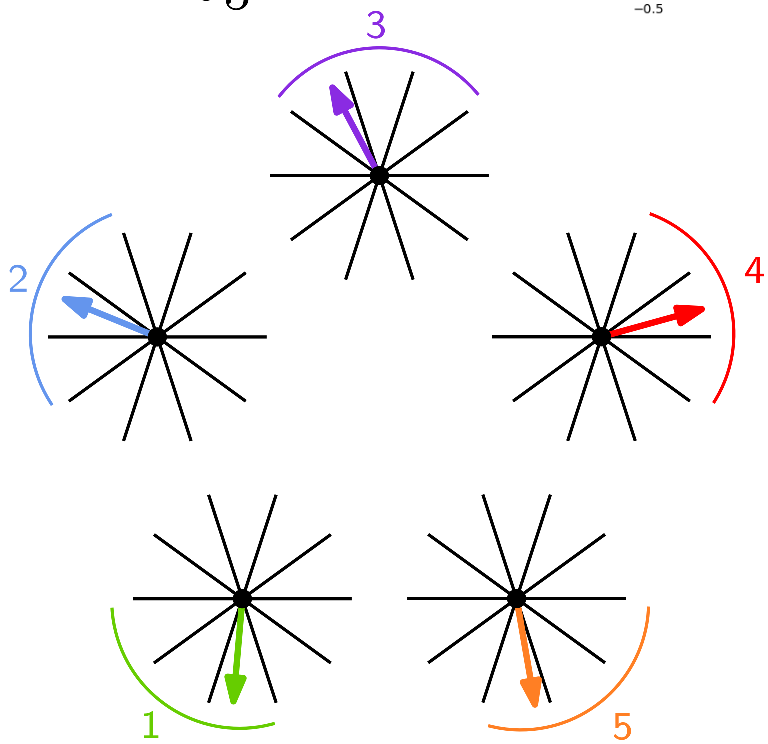
Face-counting algorithm

[Bernardi,F,Liang'23]



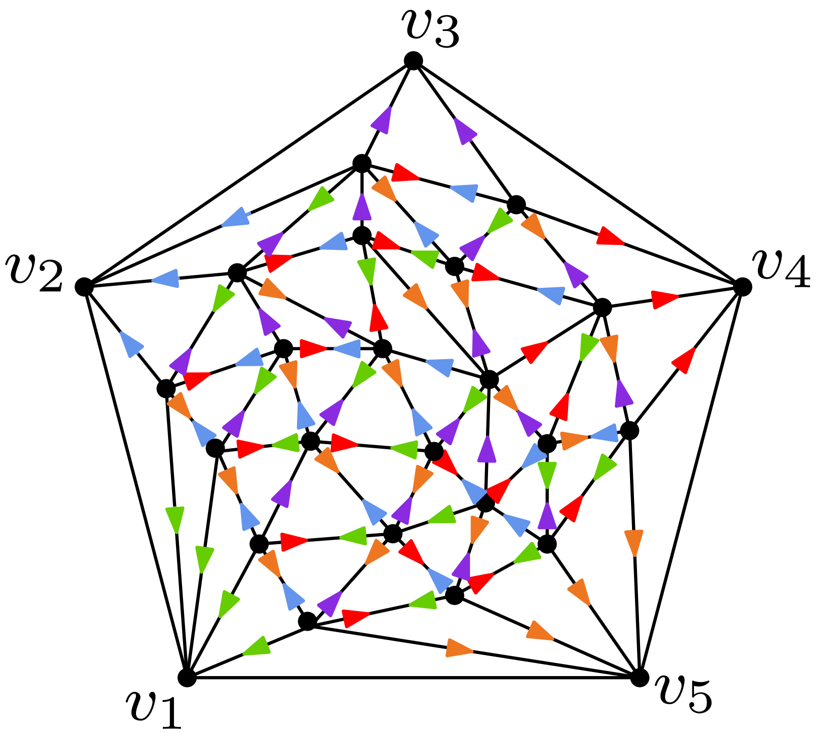


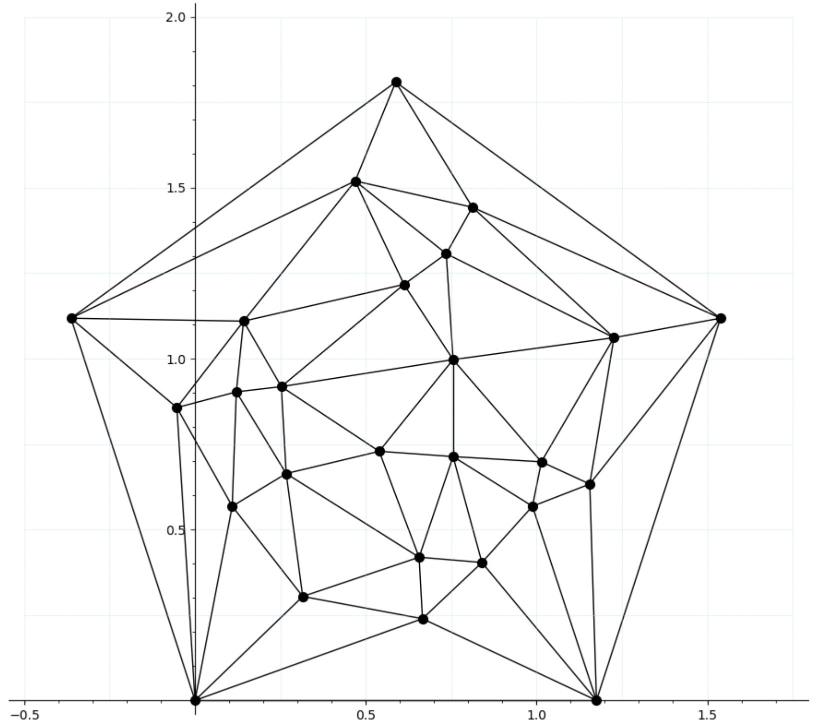
cone property



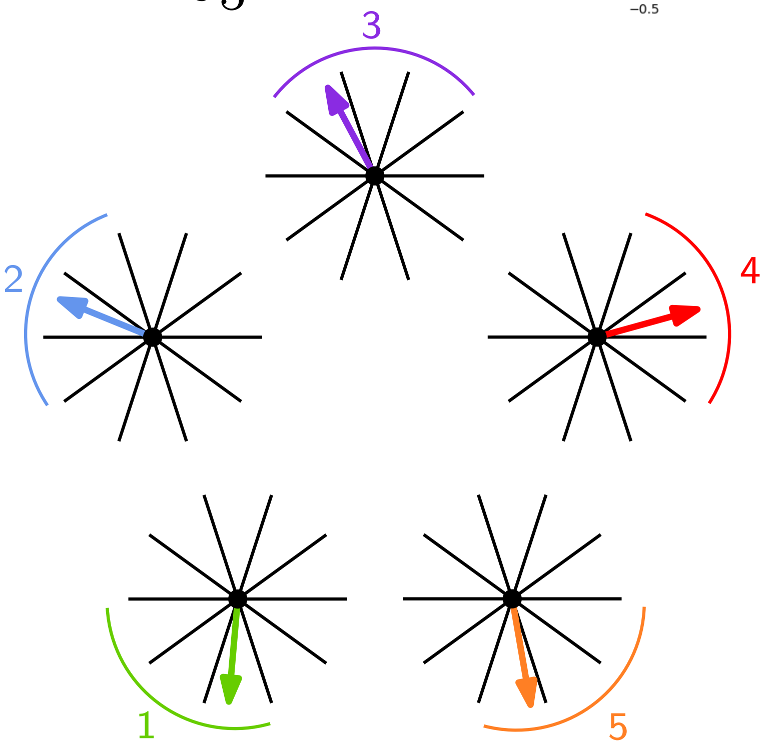
Face-counting algorithm

[Bernardi,F,Liang'23]



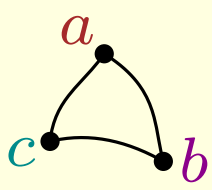


cone property

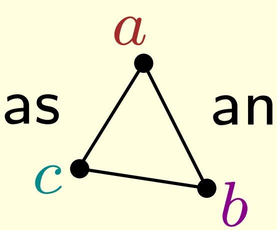


implies planarity

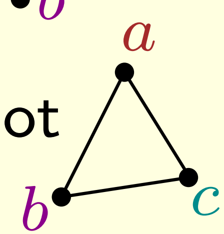
each face



drawn as

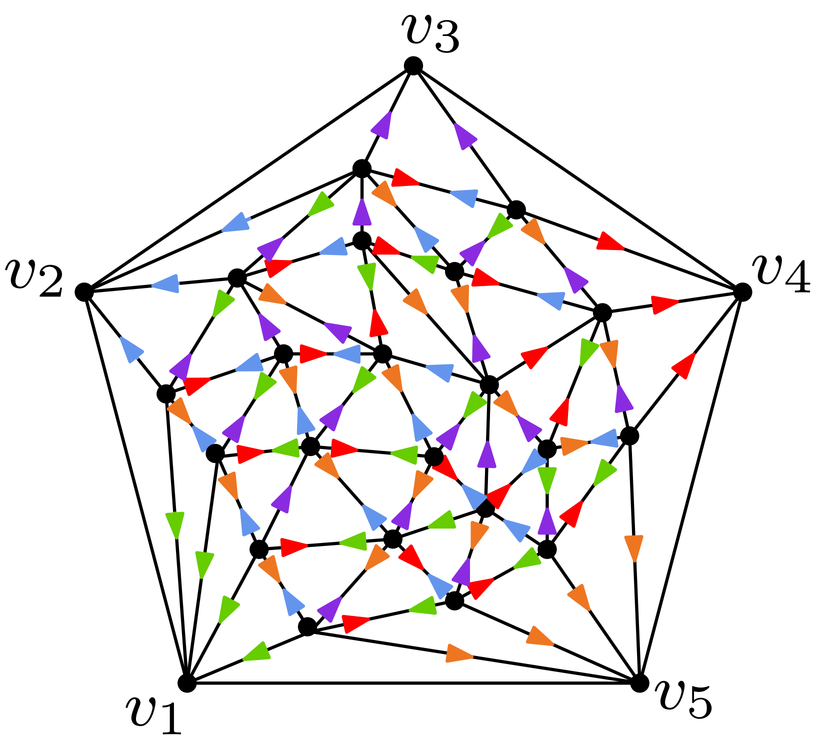


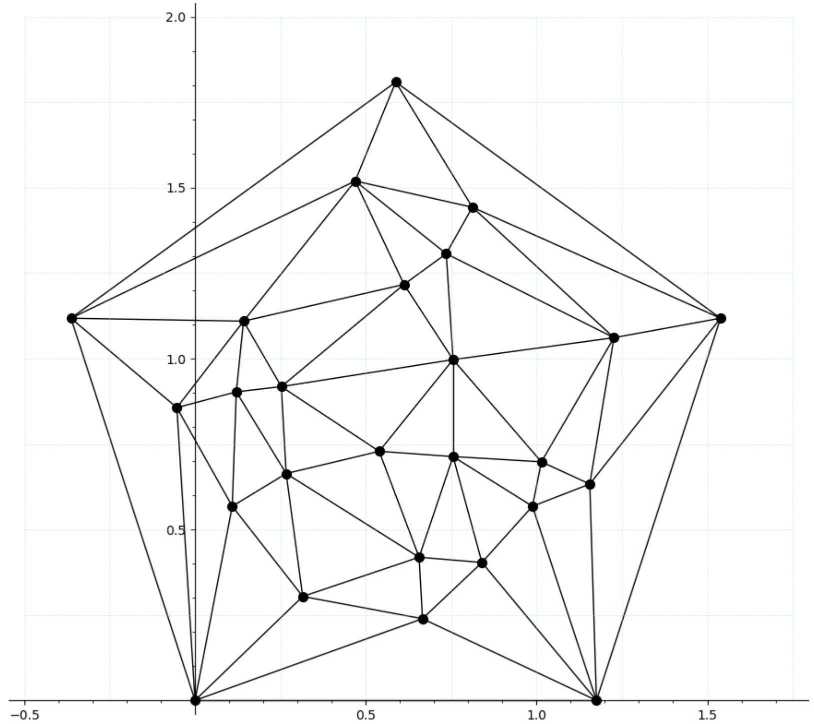
and not



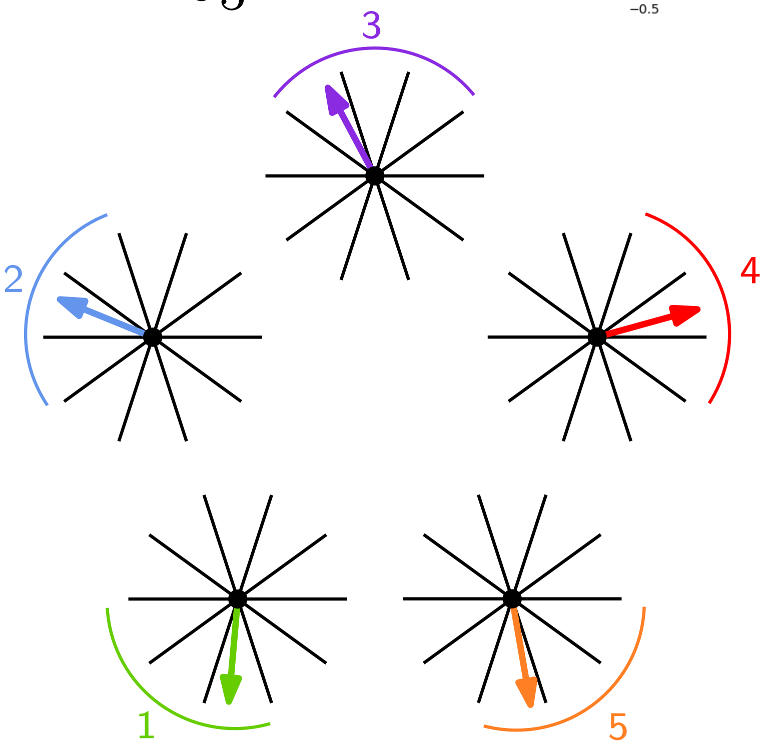
Face-counting algorithm

[Bernardi,F,Liang'23]



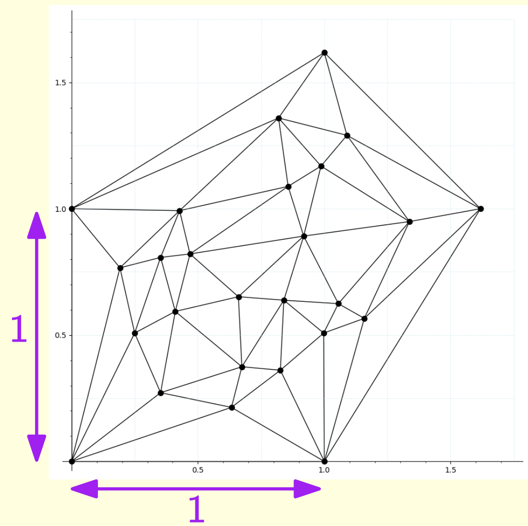


cone property



Rk: Not a grid drawing

sheer



coordinates in $\mathbb{Q}(\sqrt{5})$

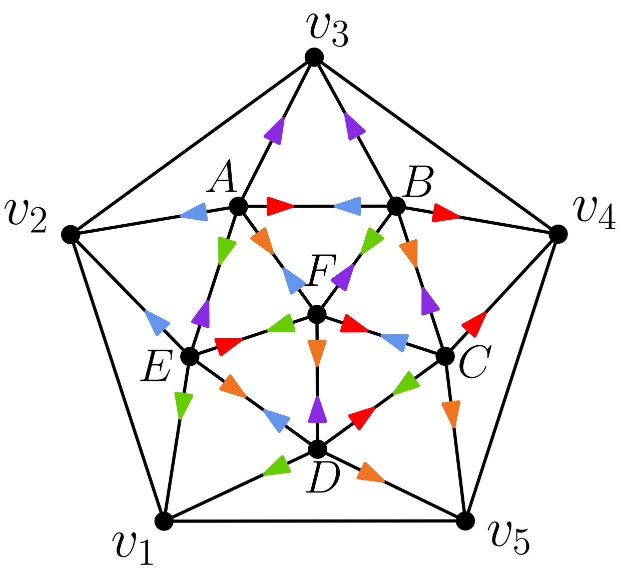
Properties and variations

- Linear time complexity

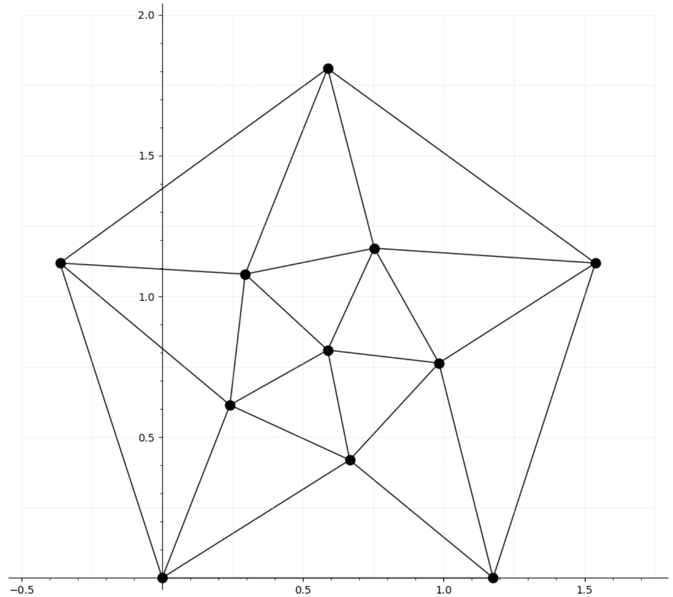
Properties and variations

- Linear time complexity

- Displays rotational symmetries



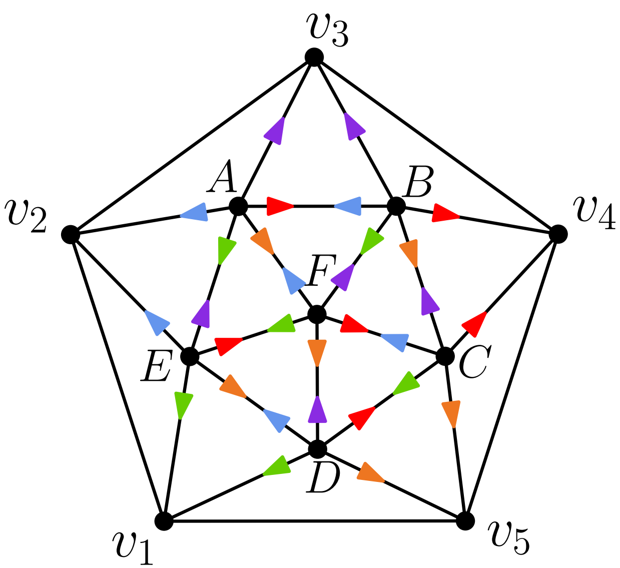
A: (2,6,4,2,1)
B: (1,2,6,4,2)
C: (2,1,2,6,4)
D: (4,2,1,2,6)
E: (6,4,2,1,2)
F: (3,3,3,3,3)



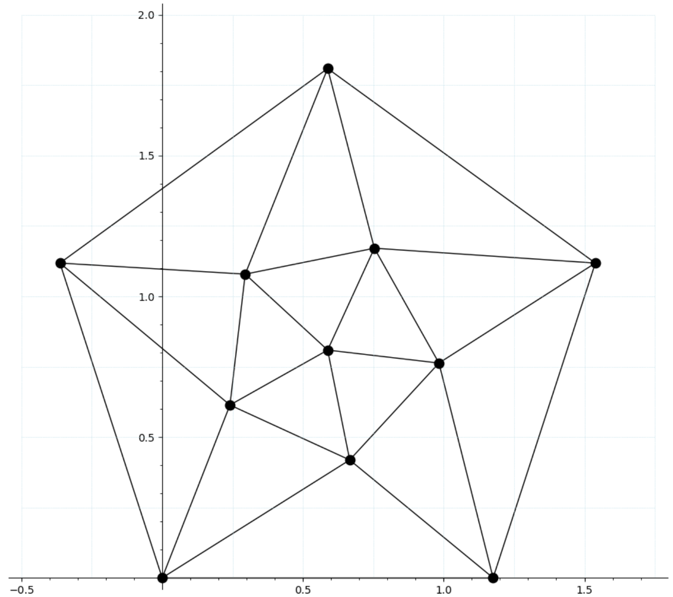
Properties and variations

- Linear time complexity

- Displays rotational symmetries



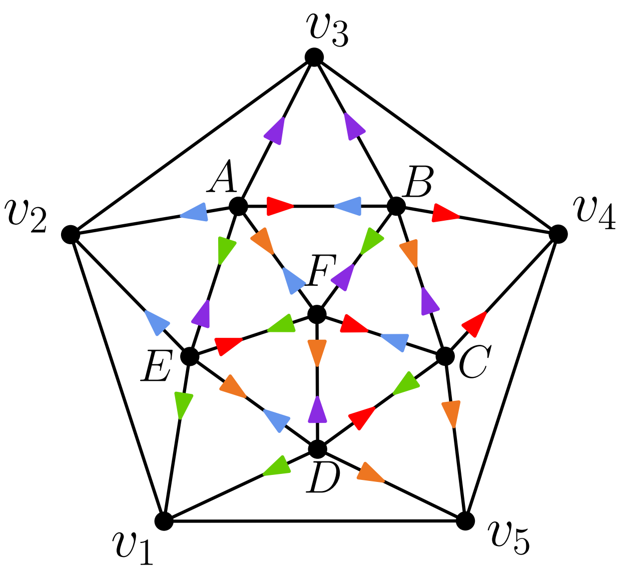
A: (2,6,4,2,1)
B: (1,2,6,4,2)
C: (2,1,2,6,4)
D: (4,2,1,2,6)
E: (6,4,2,1,2)
F: (3,3,3,3,3)



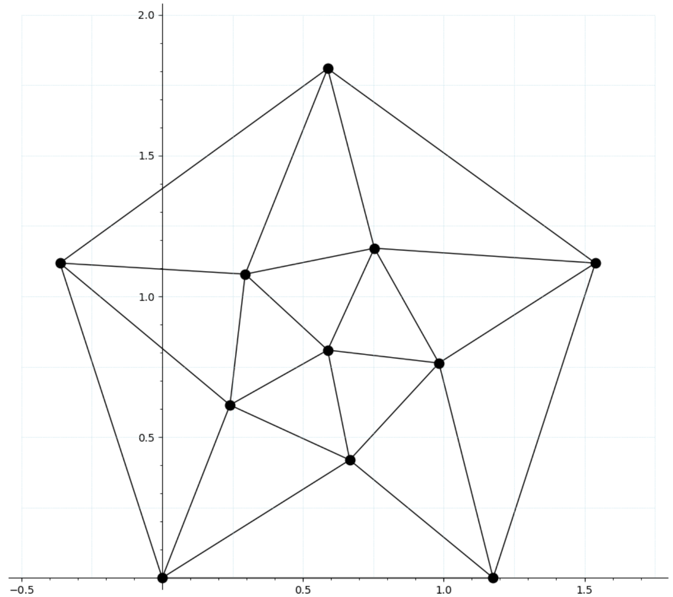
- Variations: weighted faces, vertex-counting

Properties and variations

- Linear time complexity
- Displays rotational symmetries



A: (2,6,4,2,1)
B: (1,2,6,4,2)
C: (2,1,2,6,4)
D: (4,2,1,2,6)
E: (6,4,2,1,2)
F: (3,3,3,3,3)



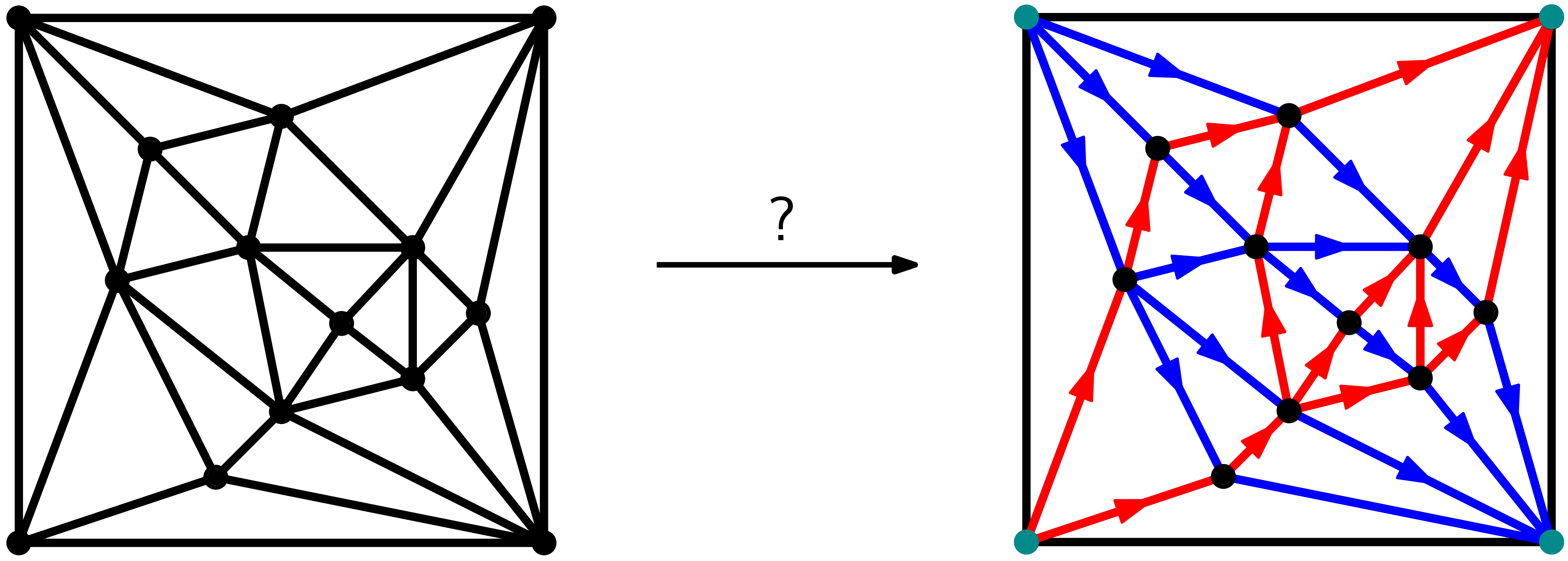
- Variations: weighted faces, vertex-counting
- Vertex resolution better than in the 3- or 4-connected drawings

smallest distance between vertices

(drawing normalized to have outer k-gon inscribed in circle of radius 1)

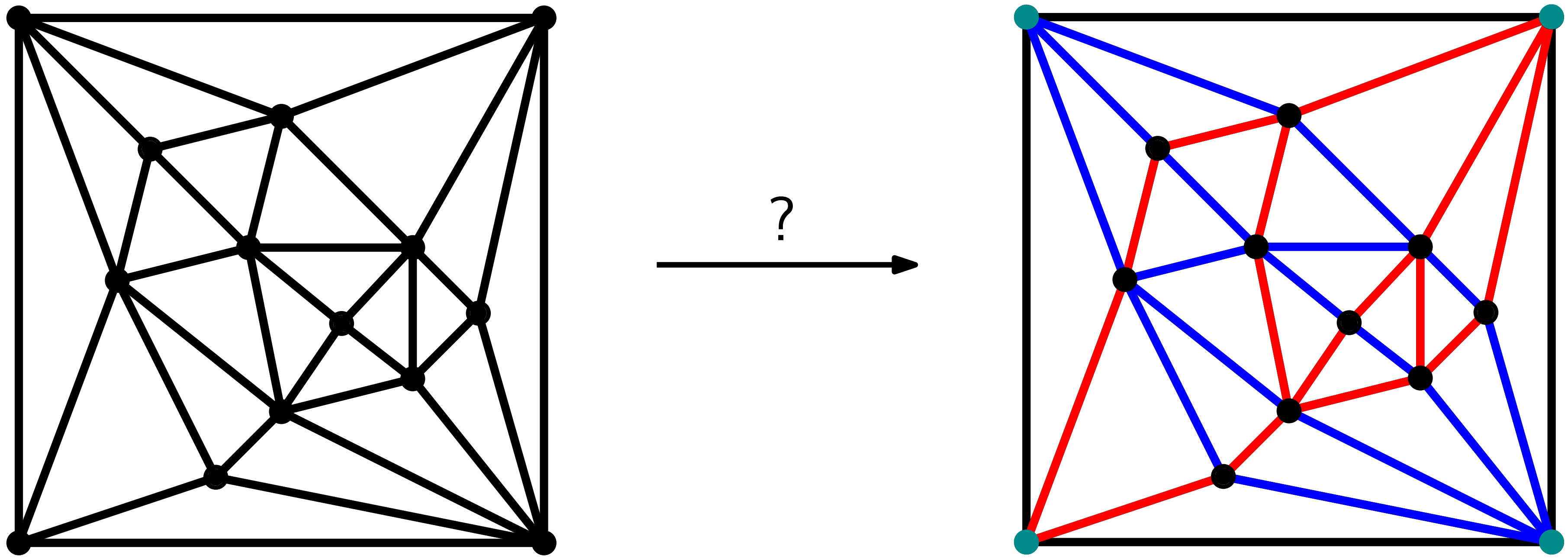
Strategy for proof of existence

new proof of existence for 4-connected (from 3-connected)



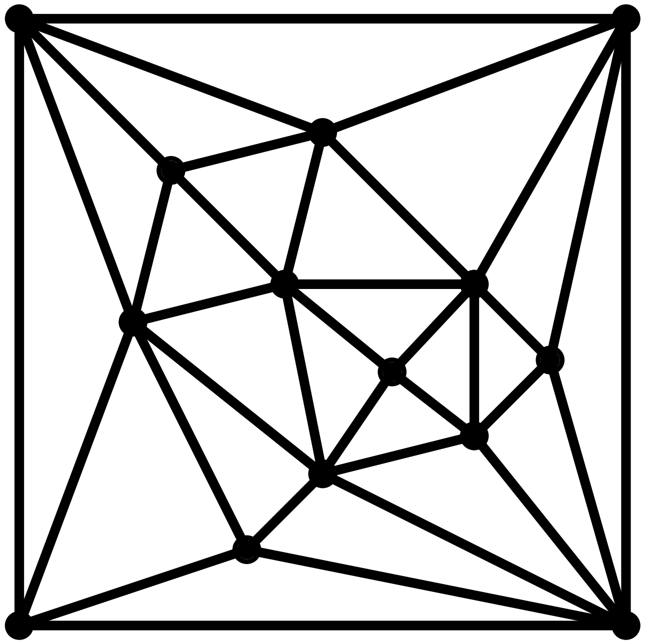
Strategy for proof of existence

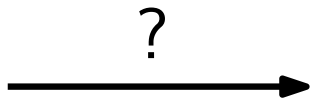
new proof of existence for 4-connected (from 3-connected)

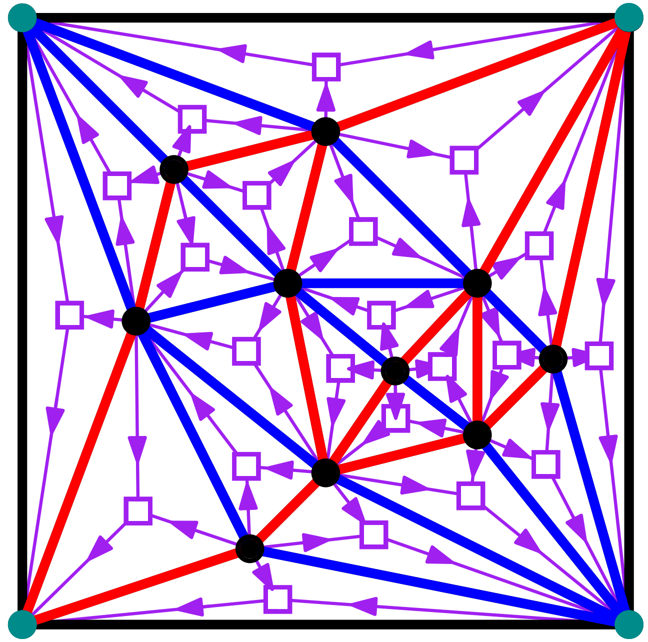


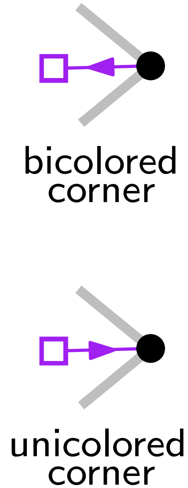
Strategy for proof of existence

new proof of existence for 4-connected (from 3-connected)



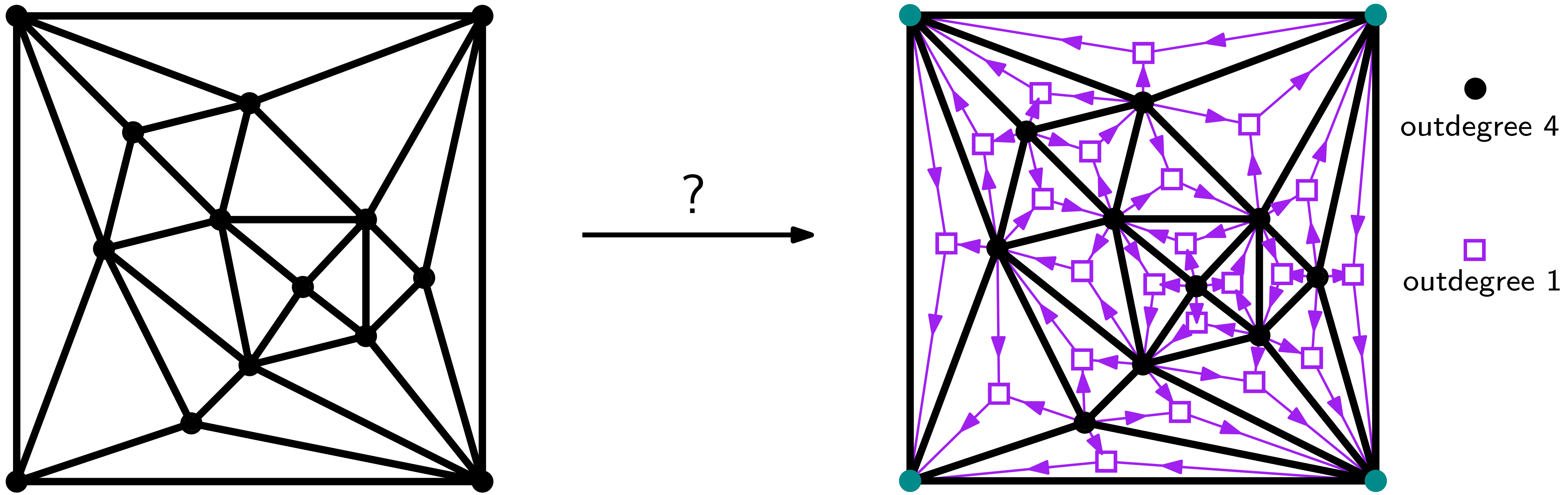






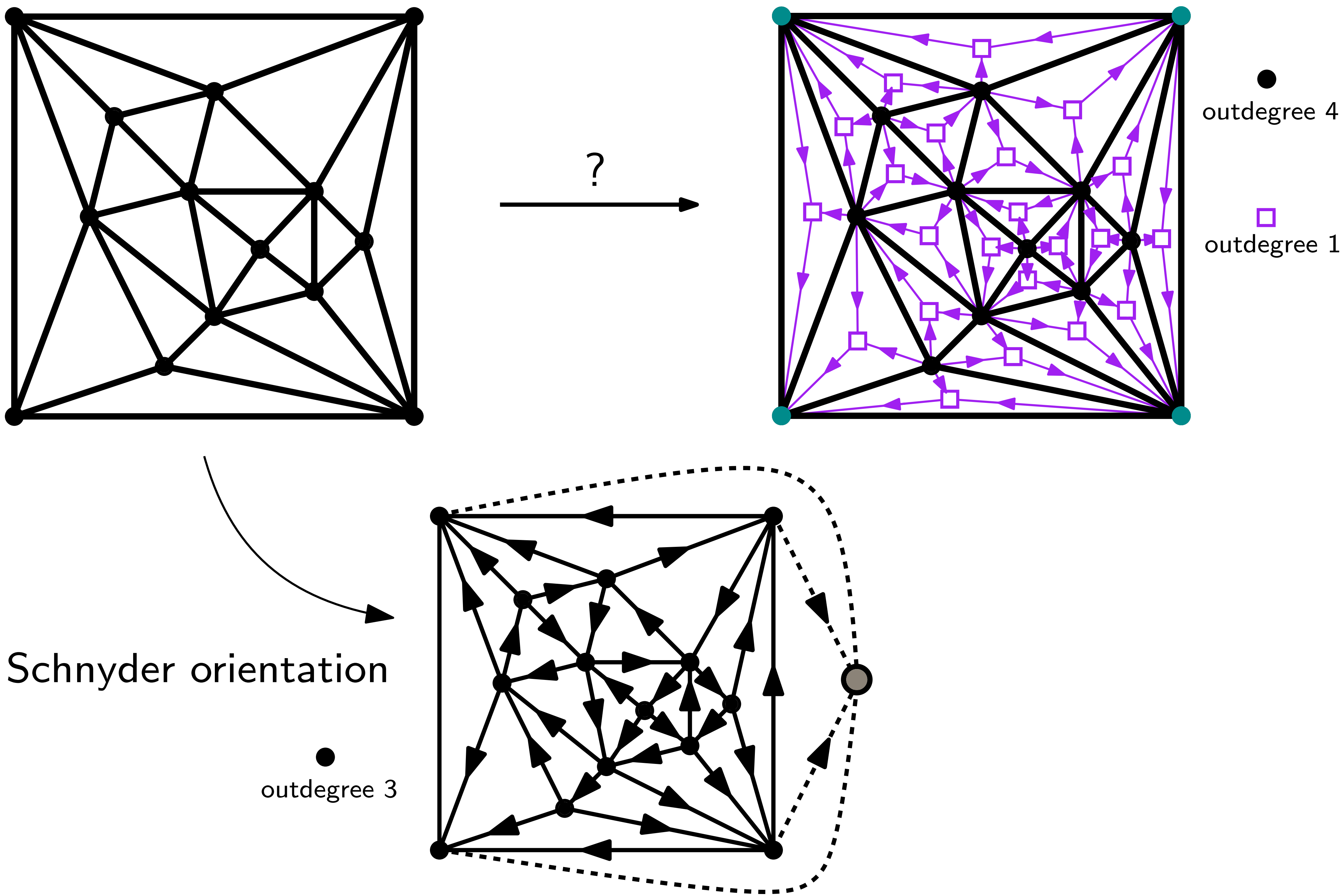
Strategy for proof of existence

new proof of existence for 4-connected (from 3-connected)



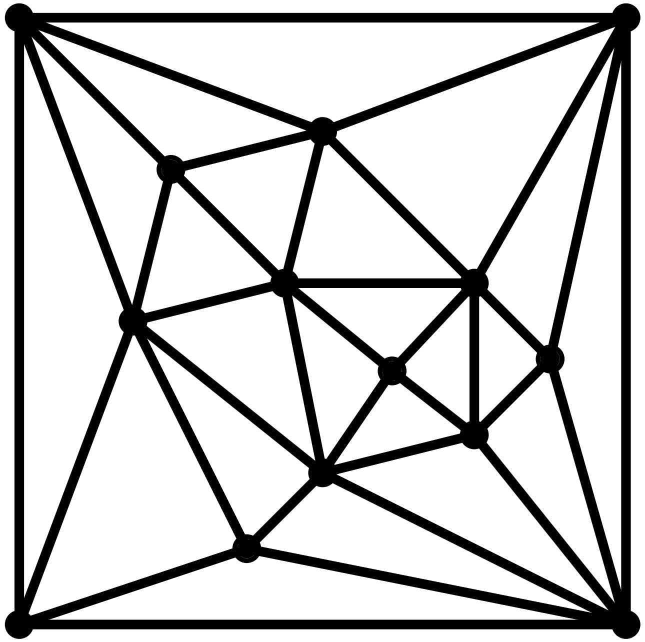
Strategy for proof of existence

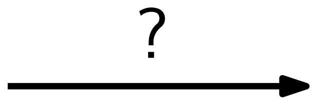
new proof of existence for 4-connected (from 3-connected)

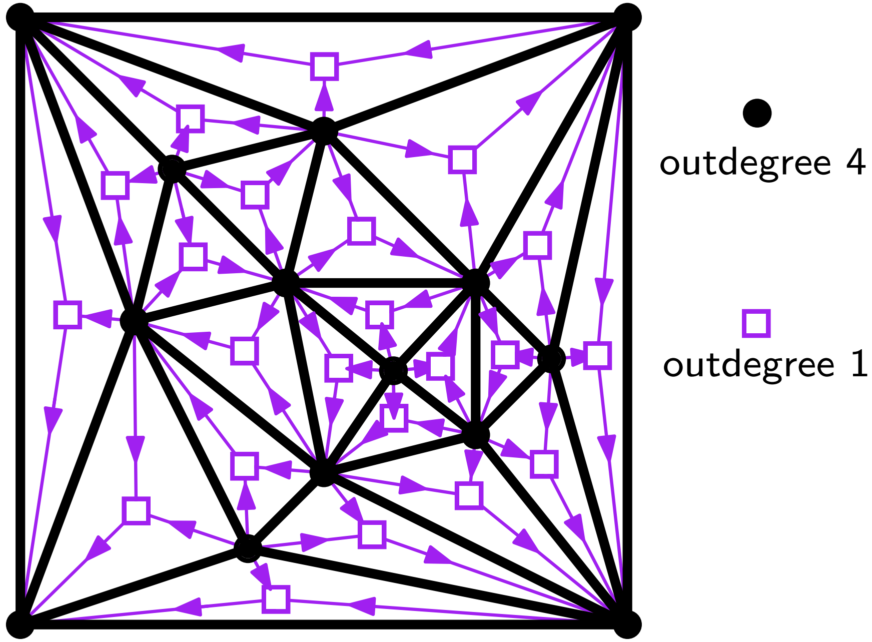


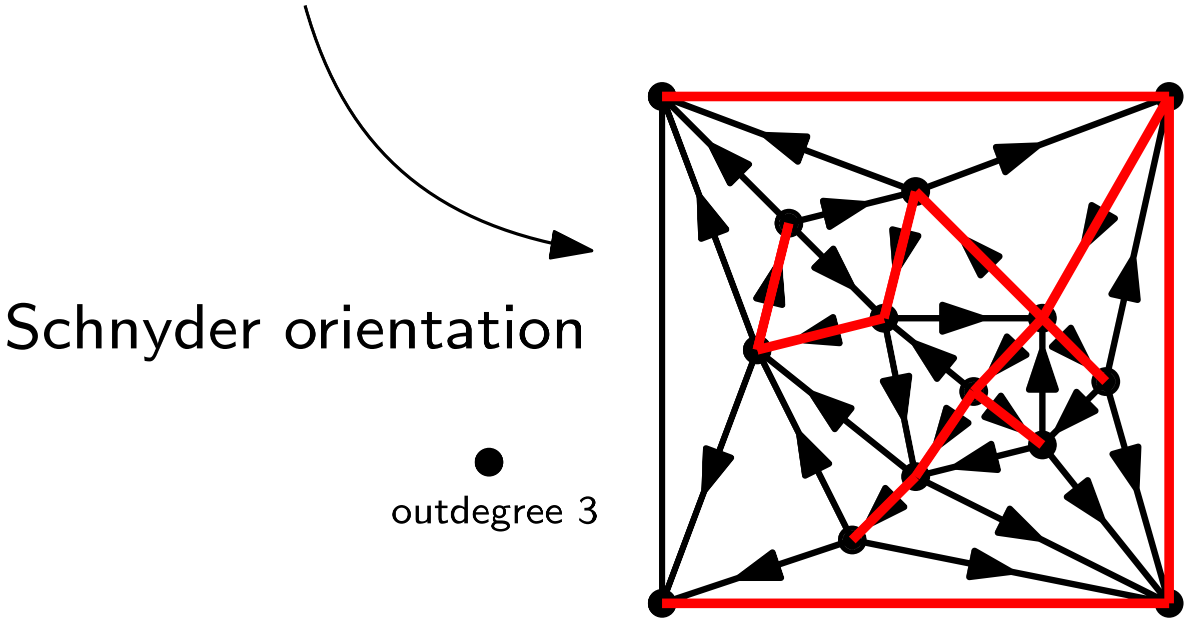
Strategy for proof of existence

new proof of existence for 4-connected (from 3-connected)









no separating triangle

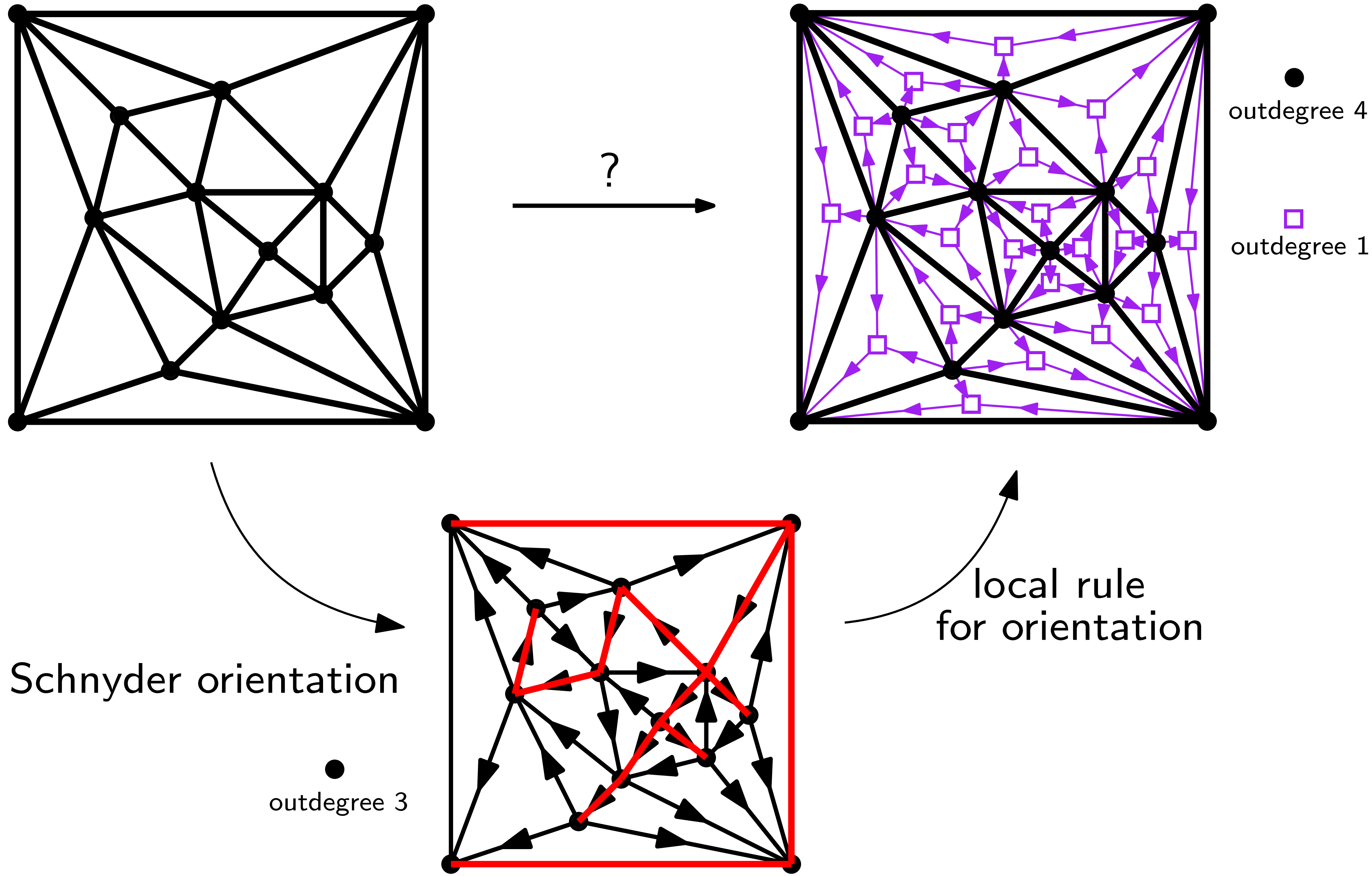
⇕

orientation is “co-accessible”

(∃ co-accessibility spanning tree)

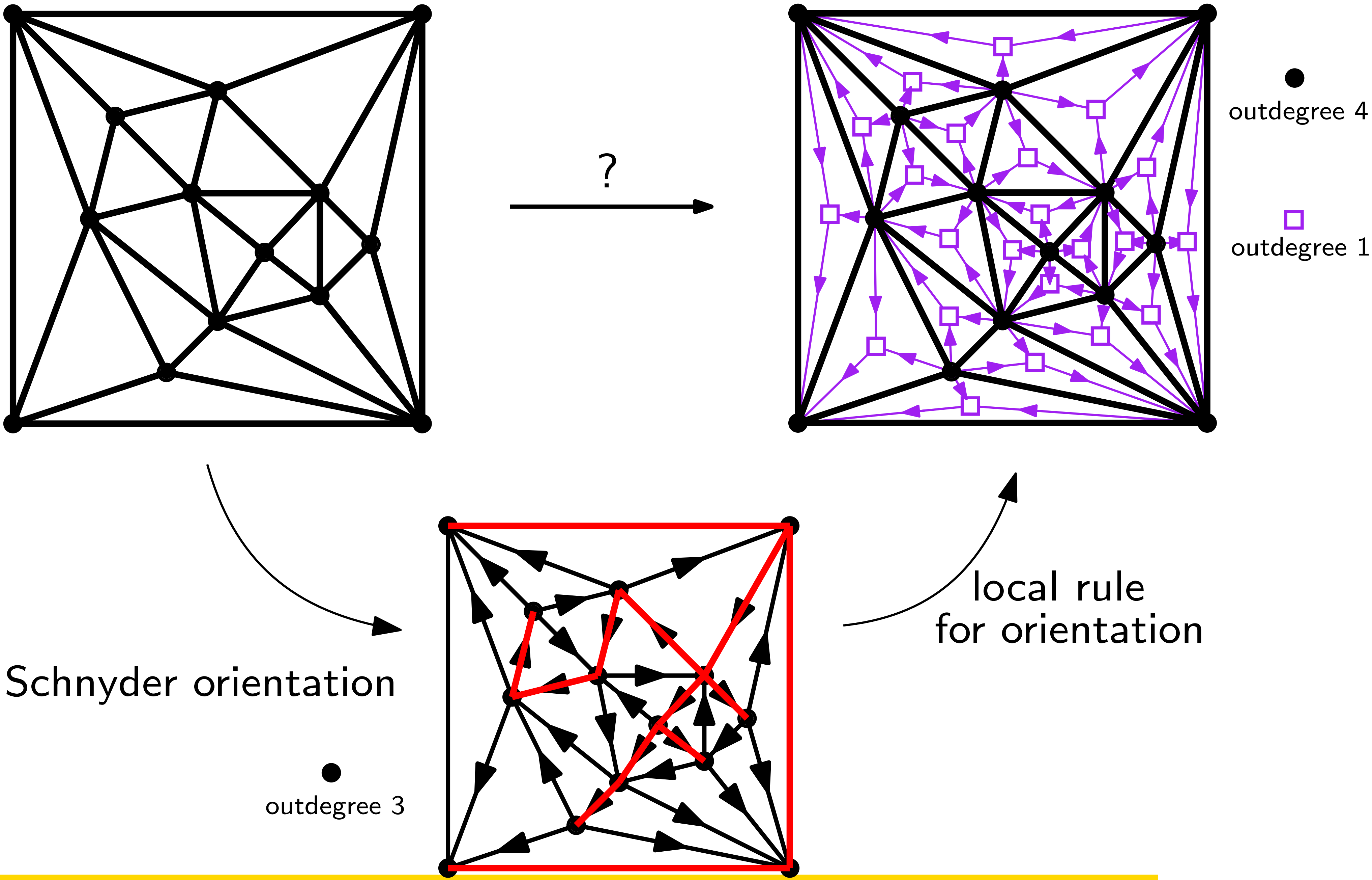
Strategy for proof of existence

new proof of existence for 4-connected (from 3-connected)



Strategy for proof of existence

new proof of existence for 4-connected (from 3-connected)



similar proof of existence for 5-connected (from 4-connected)