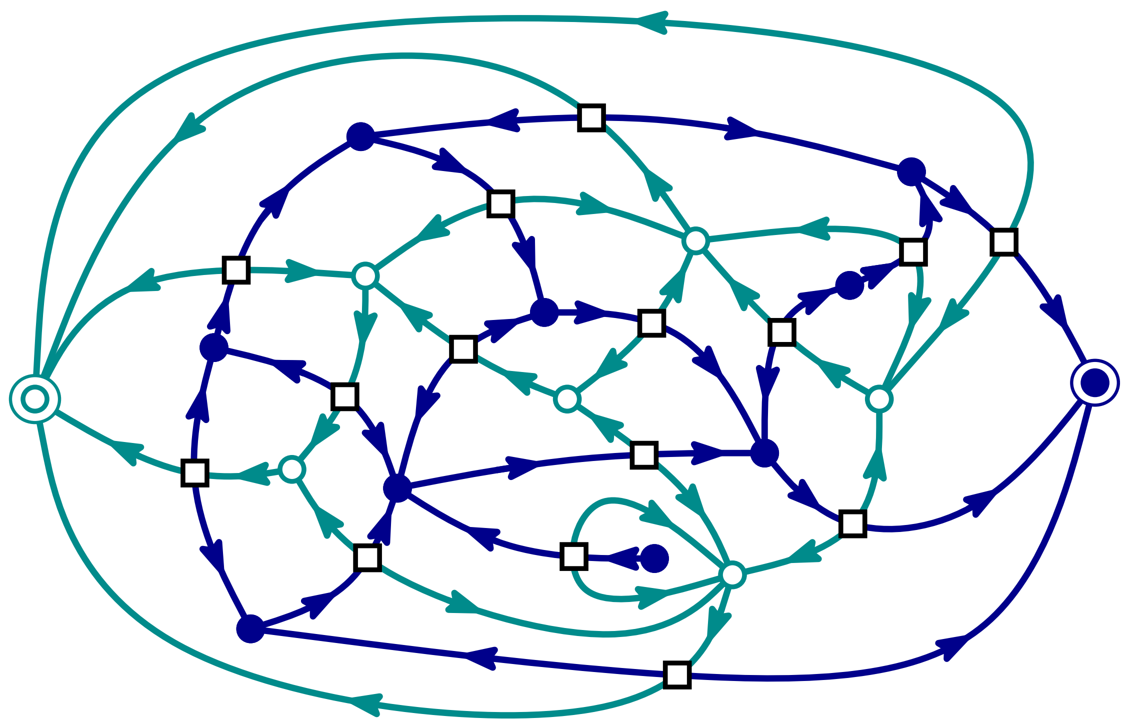
AN INTRODUCTION TO ORIENTATIONS ON MAPS

2nd lecture — May, 16th 2017

Marie Albenque (CNRS, LIX, École Polytechnique)



Mini-school on Random Maps and the Gaussian Free Field — ProbabLyon

Plan

Yesterday : Construction of orientations, existence, uniqueness

1 - Some definitions : maps, orientations.
2 - Existence of orientations
3 - Flip and flop : the lattice of orientations

Today : Applications : graph drawings, couplings, bijections

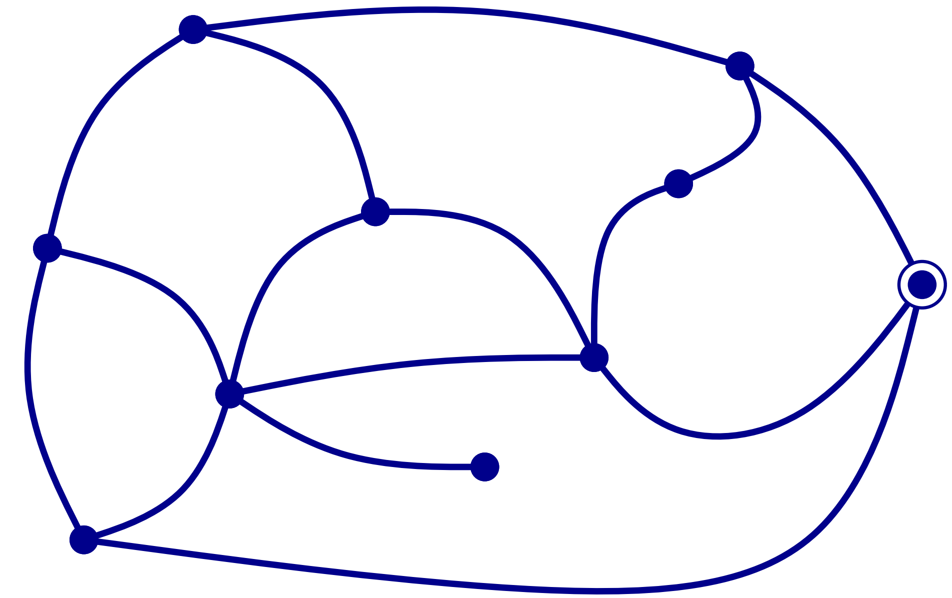
1 - Couplings and spanning trees.
2 - Schnyder woods and graph drawings.
3 - Orientations and blossoming trees

Thursday : Why should you care ?

1 - Higher genus
2 - Scaling limits for simple triangulations.
3 - Scaling limits for maps with an orientation ?

Spanning trees and orientations [Propp '93]

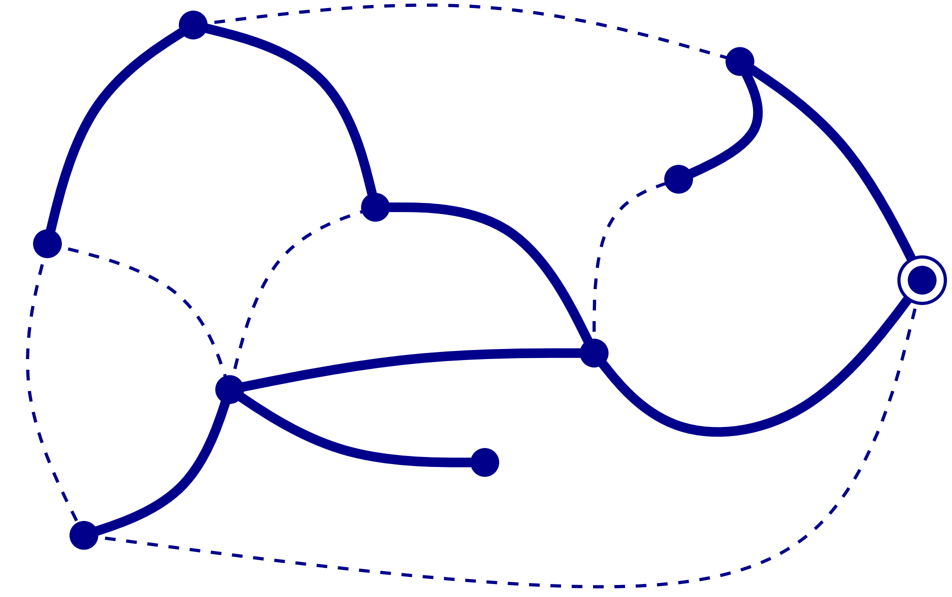
A rooted plane map M



Spanning trees and orientations [Propp '93]

A rooted plane map M

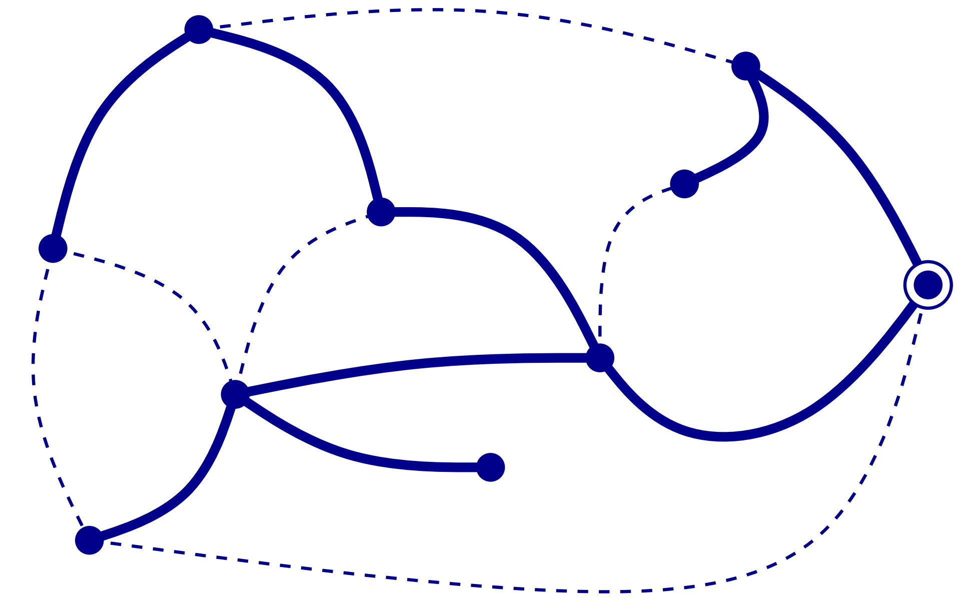
A spanning tree T



Spanning trees and orientations [Propp '93]

A rooted plane map M

A spanning tree T



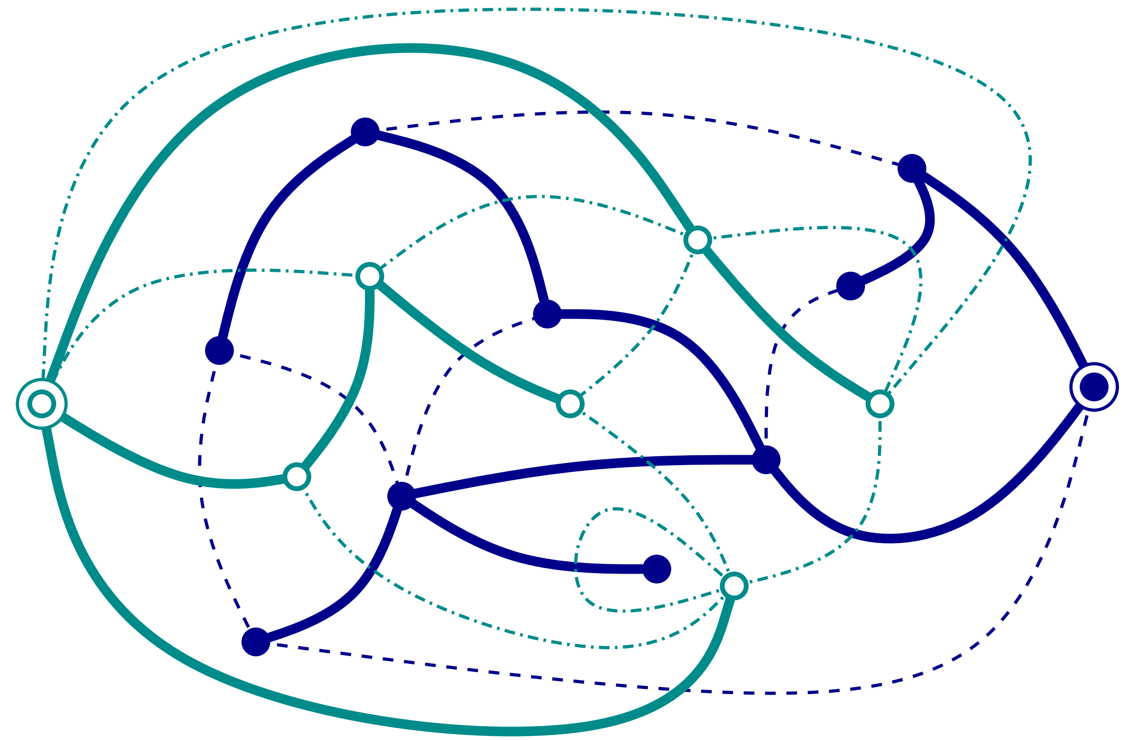
Goal : For a fixed M, what can we say about the structure of spanning trees ? Can we endow them with an orientation structure ?

Spanning trees and orientations [Propp '93]

A rooted plane map M

A spanning tree T

$T^\star$ = spanning tree of $M^\star$



Convention :

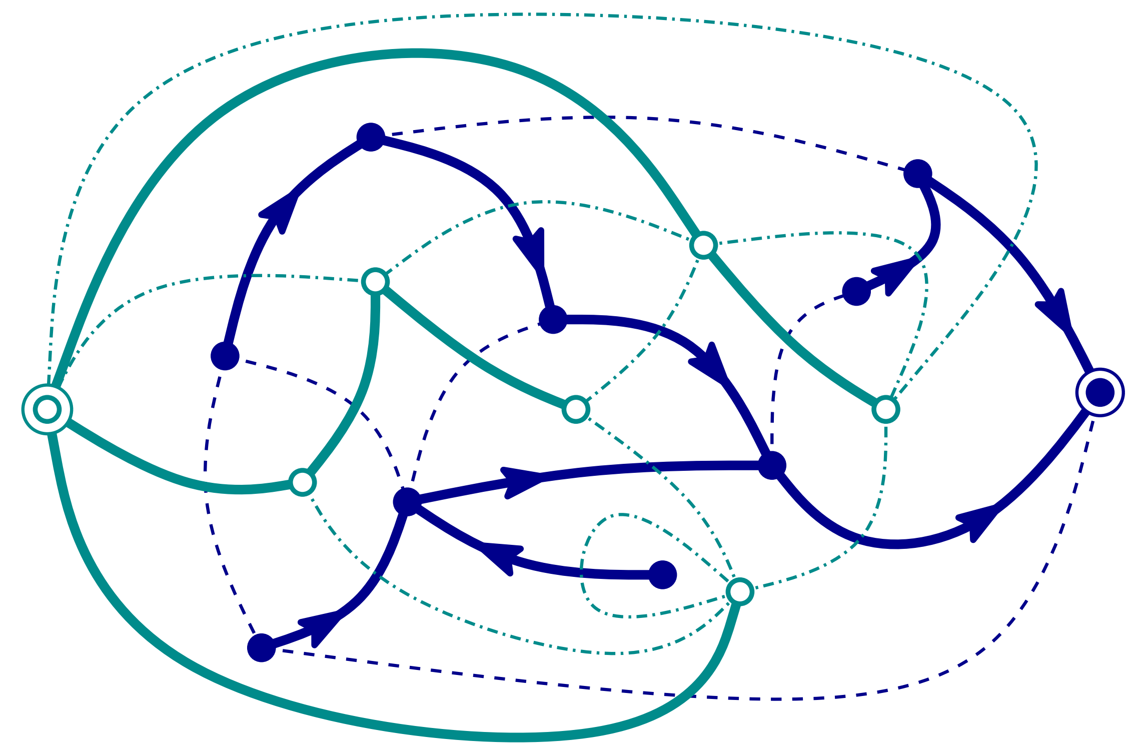
Goal : For a fixed M, what can we say about the structure of spanning trees ? Can we endow them with an orientation structure ?

Spanning trees and orientations [Propp '93]

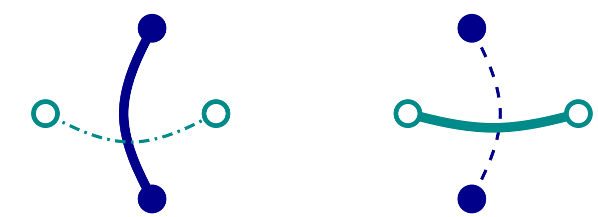
A rooted plane map M

A spanning tree T

$T^\star$ = spanning tree of $M^\star$



Convention :



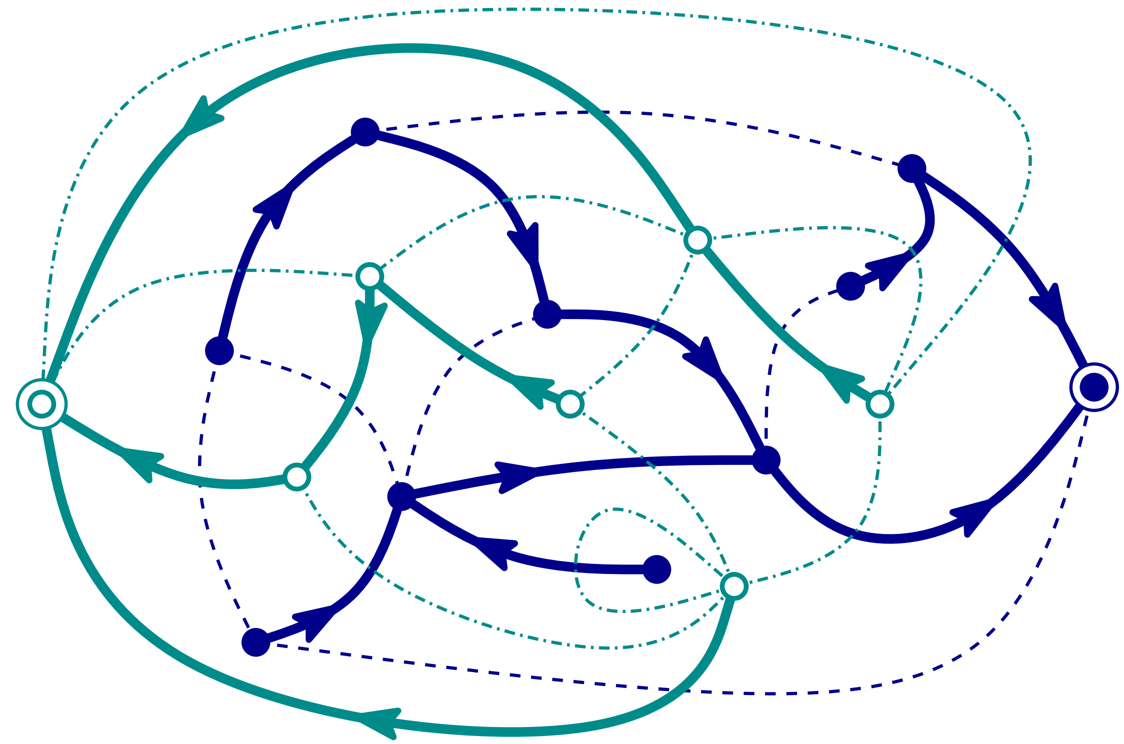
Goal : For a fixed M, what can we say about the structure of spanning trees ? Can we endow them with an orientation structure ?

Spanning trees and orientations [Propp '93]

A rooted plane map M

A spanning tree T

$T^\star$ = spanning tree of $M^\star$

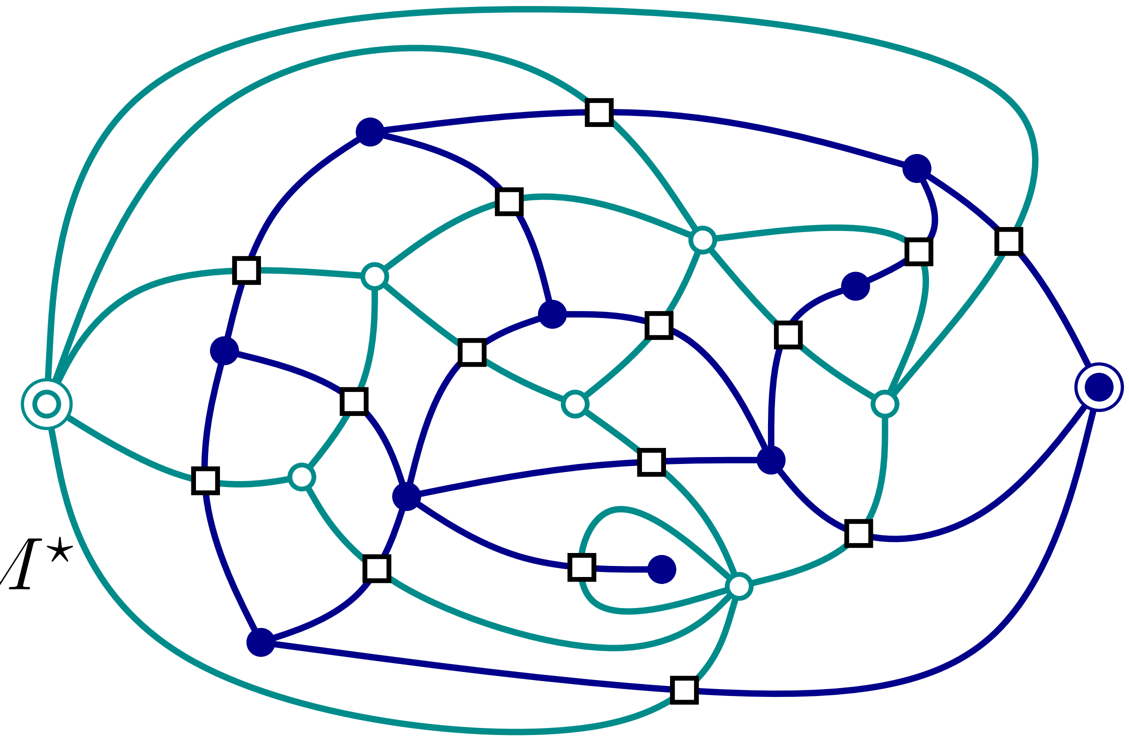


Convention :

Goal : For a fixed M, what can we say about the structure of spanning trees ? Can we endow them with an orientation structure ?

Spanning trees and orientations [Propp '93]

A rooted plane map M



$\tilde{M}$ = **completion** of M

= superimposition of M and $M^\star$

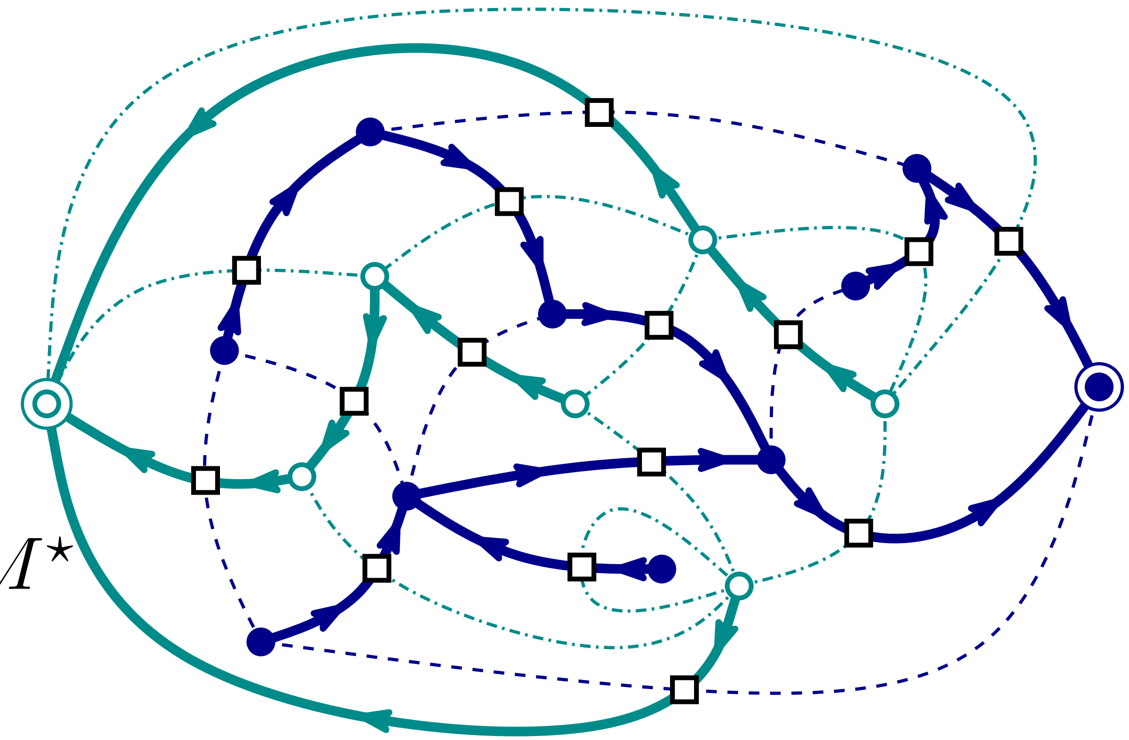
Spanning trees and orientations [Propp '93]

A rooted plane map M

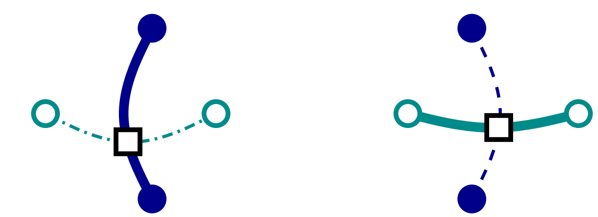
A spanning tree T

$T^\star$ = spanning tree of $M^\star$

$\tilde{M}$ = **completion** of M
= superimposition of M and $M^\star$



Convention :



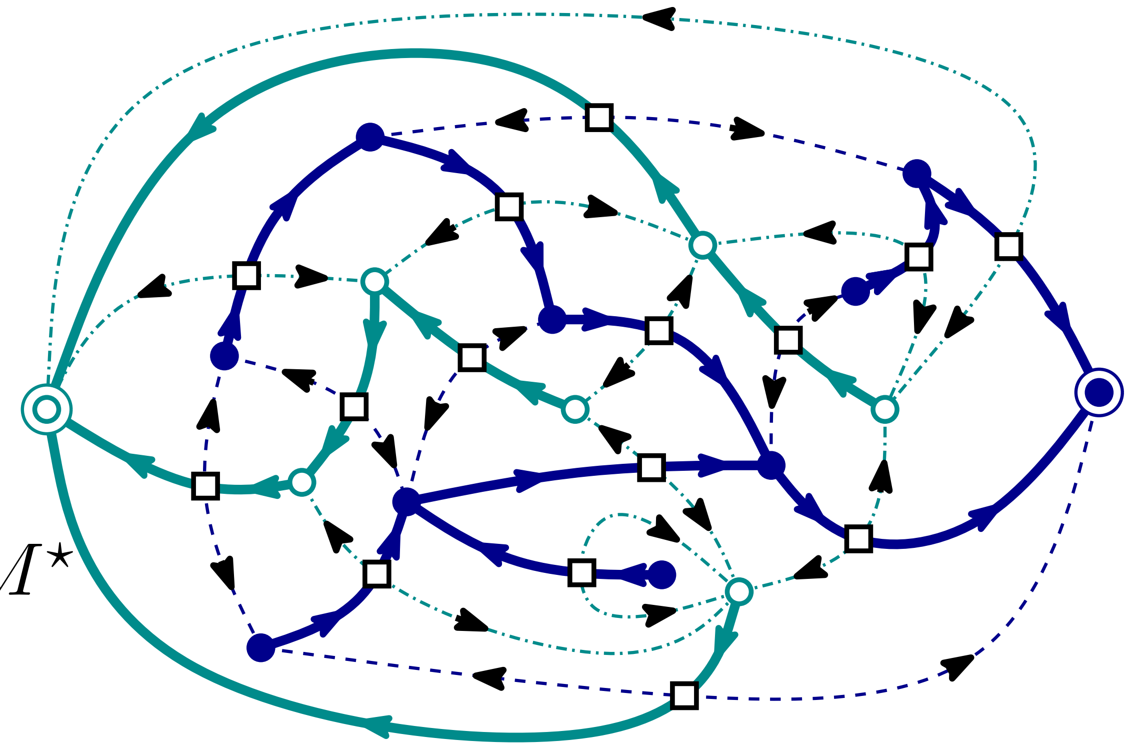
Spanning trees and orientations [Propp '93]

A rooted plane map M

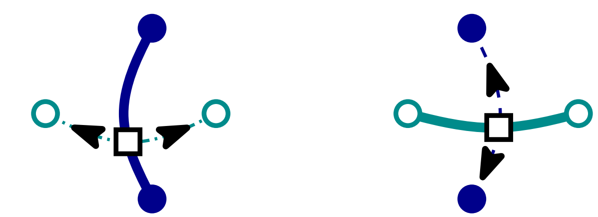
A spanning tree T

$T^\star$ = spanning tree of $M^\star$

$\tilde{M}$ = **completion** of M
= superimposition of M and $M^\star$



Convention :



For a spanning tree T of M, we define the α_T-orientation of $\tilde{M}$ by :

$$\begin{cases} \alpha_T(\circledcirc) = \alpha_T(\odot) = 0 \\ \alpha_T(\circ) = \alpha_T(\bullet) = 1 \\ \alpha_T(\square) = 3 \end{cases}$$

Spanning trees and orientations [Propp '93]

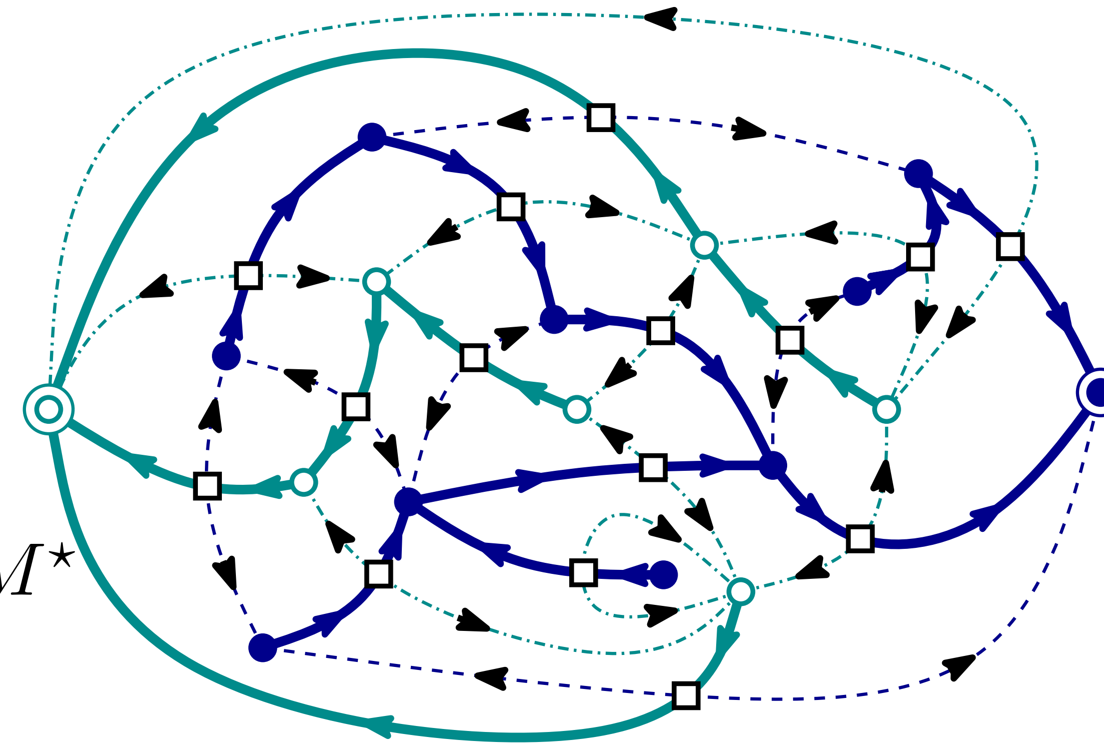
A rooted plane map M

A spanning tree T

$T^\star$ = spanning tree of $M^\star$

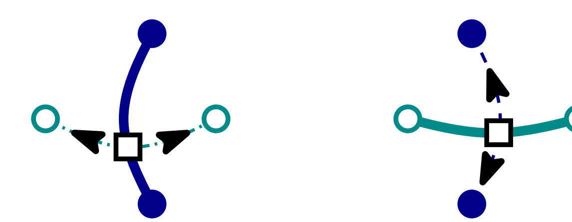
$\tilde{M}$ = **completion** of M

$\quad = $ superimposition of M and $M^\star$



For a spanning tree T of M, we define the α_T-orientation of $\tilde{M}$ by :

Convention :



$$\begin{cases} \alpha_T(\circledcirc) = \alpha_T(\odot) = 0 \\ \alpha_T(\circ) = \alpha_T(\bullet) = 1 \\ \alpha_T(\square) = 3 \end{cases}$$

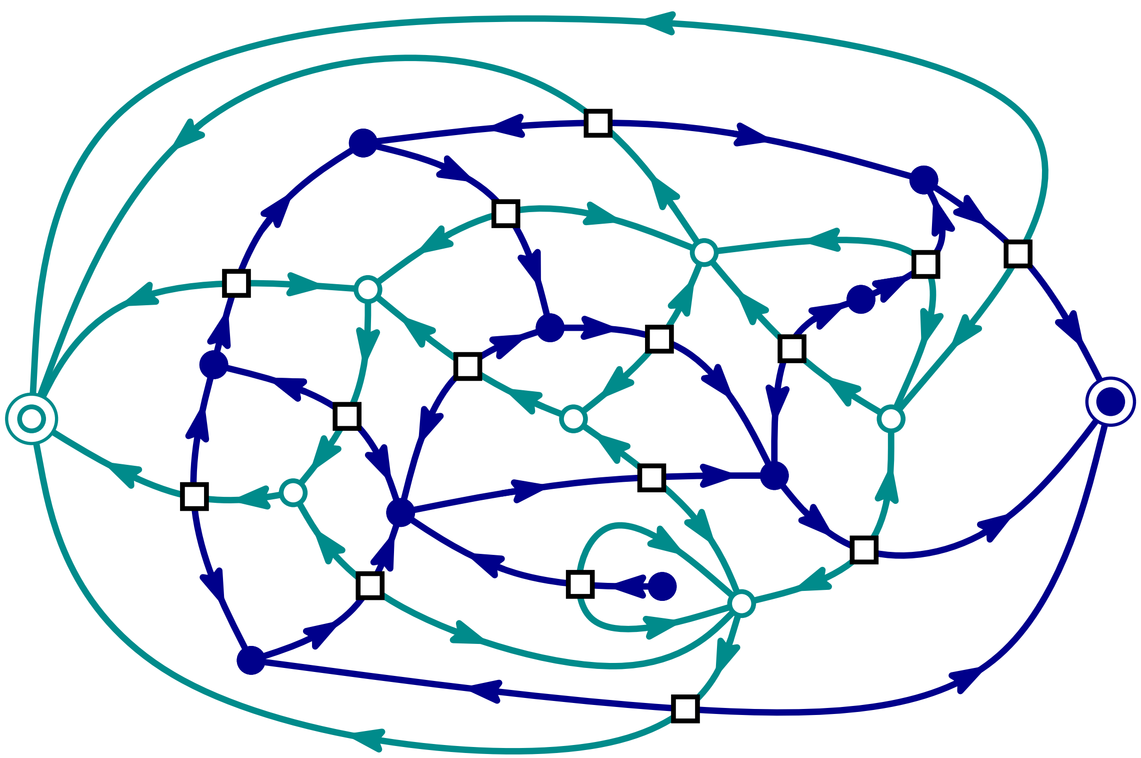
And reciprocally ?

Spanning trees and orientations

Proposition : [Propp '93, Felsner '04]

The spanning trees of M are in bijection with the α_T-orientations of $\tilde{M}$.

$$\begin{cases} \alpha_T(\circledcirc) = \alpha_T(\odot) = 0 \\ \alpha_T(\circ) = \alpha_T(\bullet) = 1 \\ \alpha_T(\square) = 3 \end{cases}$$

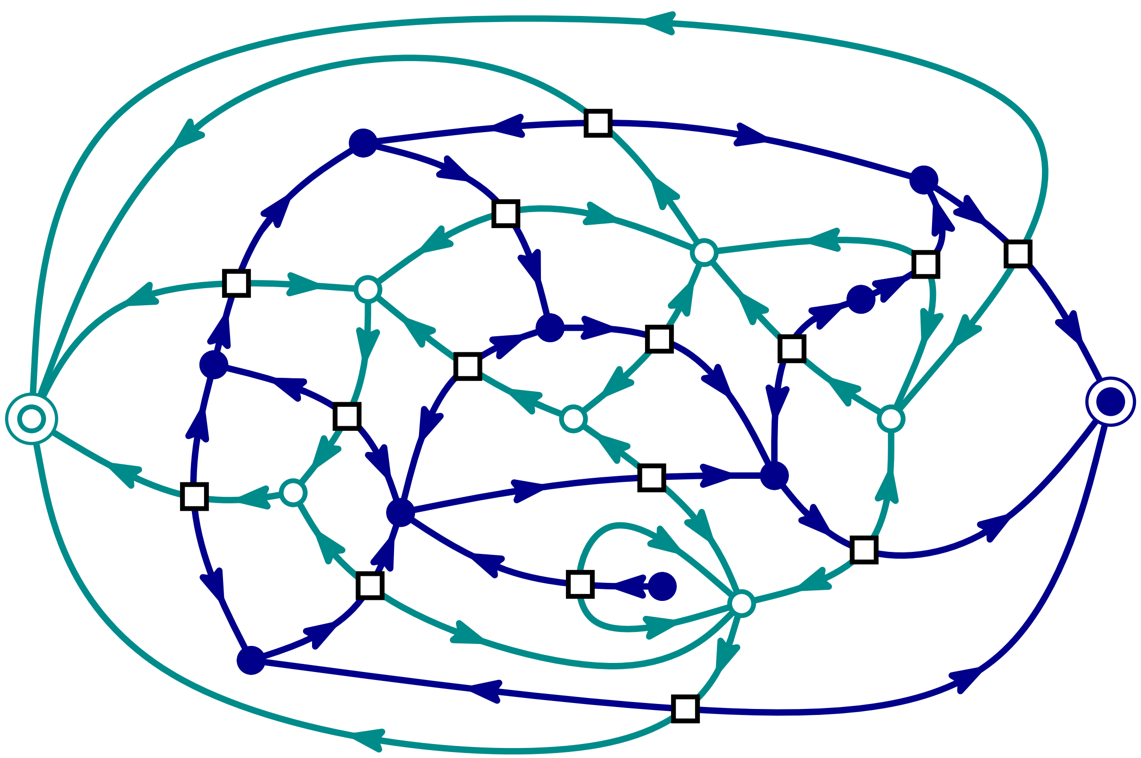


Spanning trees and orientations

Proposition : [Propp '93, Felsner '04]

The spanning trees of M are in bijection with the α_T-orientations of $\tilde{M}$.

$$\begin{cases} \alpha_T(\circledcirc) = \alpha_T(\odot) = 0 \\ \alpha_T(\circ) = \alpha_T(\bullet) = 1 \\ \alpha_T(\square) = 3 \\ \qquad\;\; = \deg(\square) - 1 \end{cases}$$

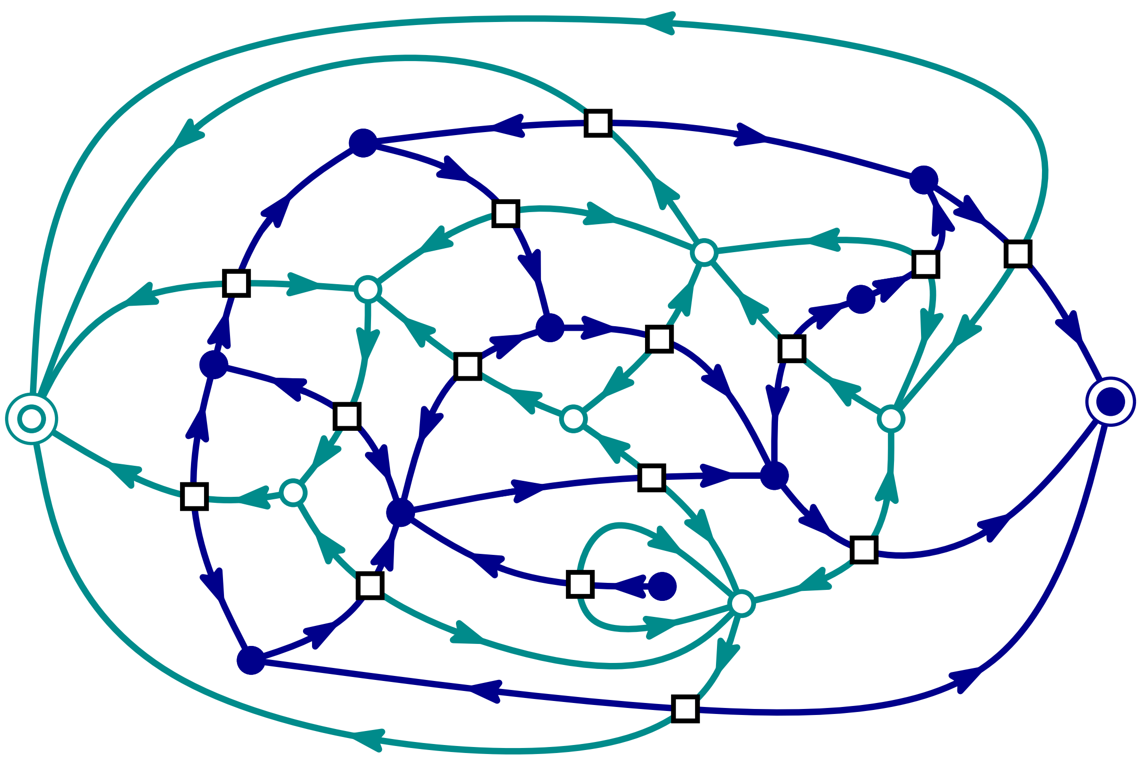


Spanning trees and orientations

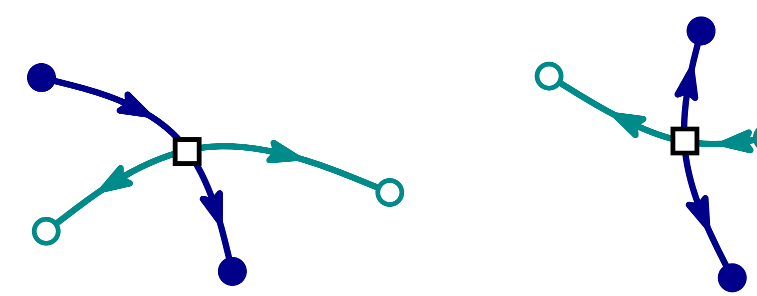
Proposition : [Propp '93, Felsner '04]

The spanning trees of M are in bijection with the α_T-orientations of $\tilde{M}$.

$$\begin{cases} \alpha_T(\circledcirc) = \alpha_T(\odot) = 0 \\ \alpha_T(\circ) = \alpha_T(\bullet) = 1 \\ \alpha_T(\square) = 3 \end{cases}$$



At each vertex $\square$, 2 possible configurations :

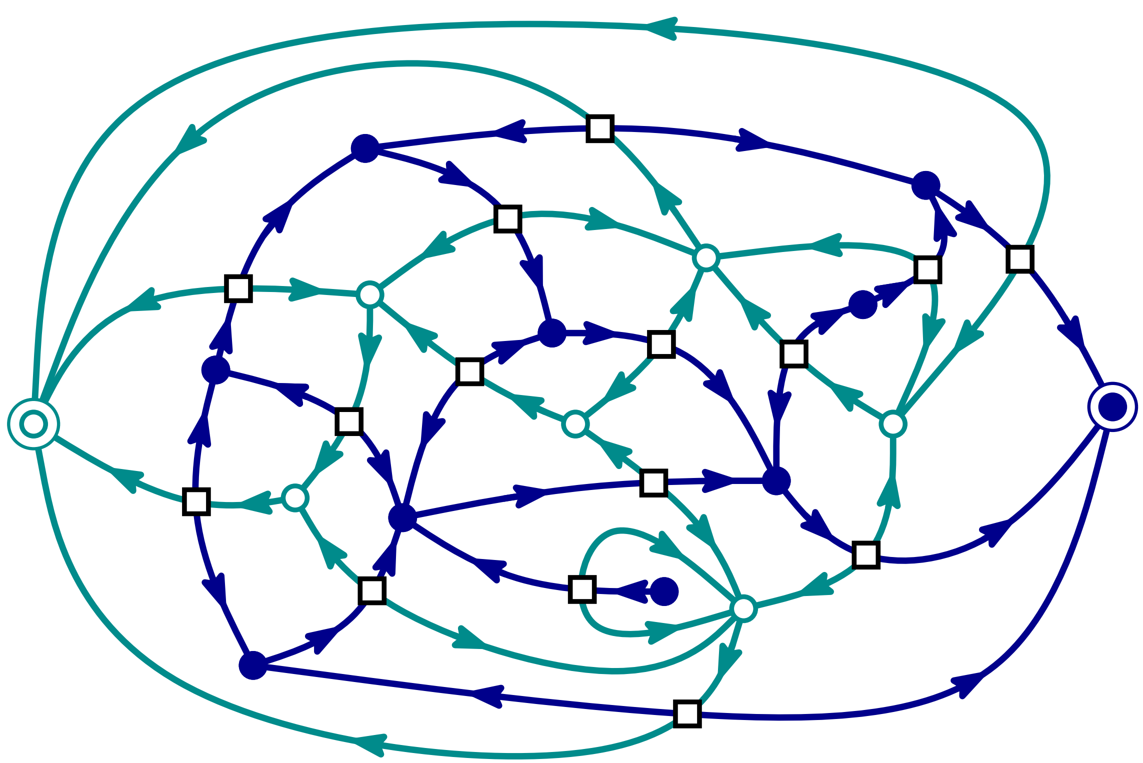


Spanning trees and orientations

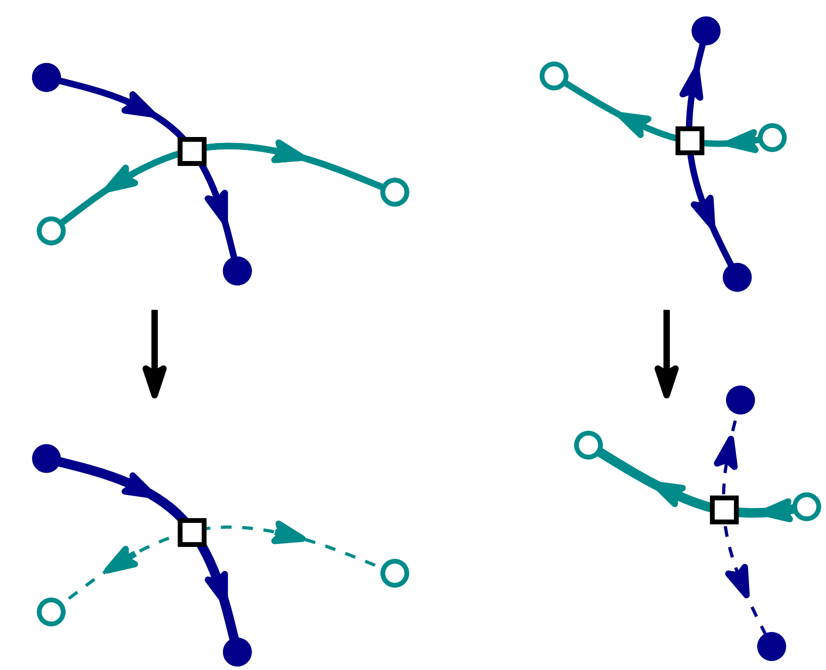
Proposition : [Propp '93, Felsner '04]

The spanning trees of M are in bijection with the α_T-orientations of $\tilde{M}$.

$$\begin{cases} \alpha_T(\circledcirc) = \alpha_T(\odot) = 0 \\ \alpha_T(\circ) = \alpha_T(\bullet) = 1 \\ \alpha_T(\square) = 3 \end{cases}$$



At each vertex $\square$, 2 possible configurations :

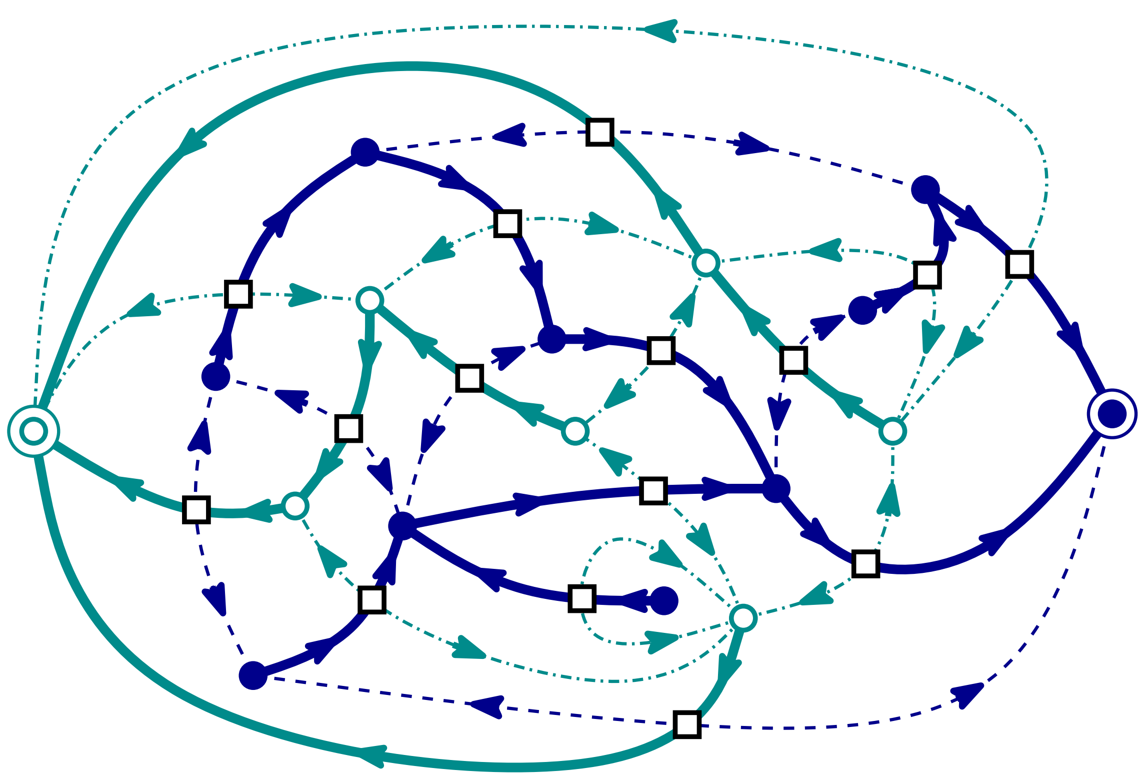


Spanning trees and orientations

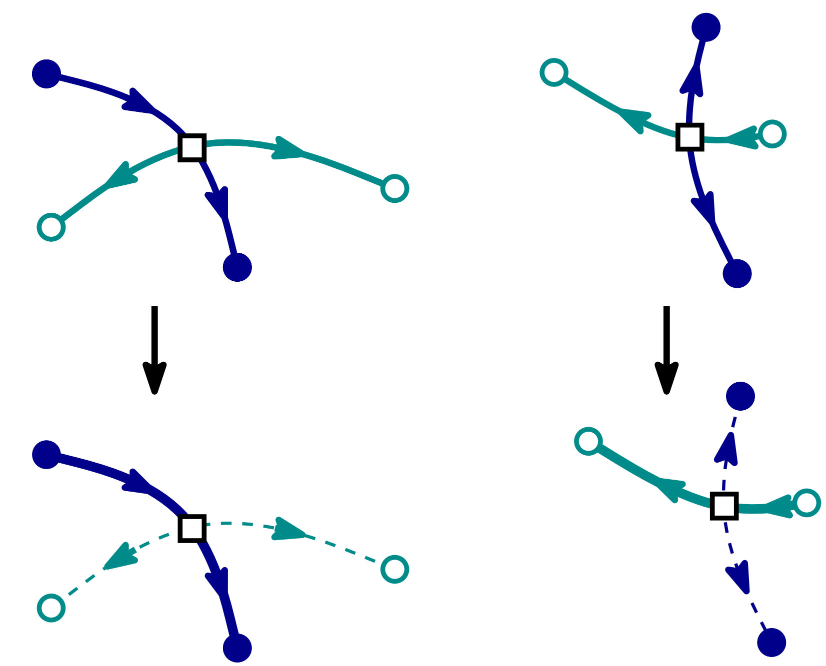
Proposition : [Propp '93, Felsner '04]

The spanning trees of M are in bijection with the α_T-orientations of $\tilde{M}$.

$$\begin{cases} \alpha_T(\circledcirc) = \alpha_T(\odot) = 0 \\ \alpha_T(\circ) = \alpha_T(\bullet) = 1 \\ \alpha_T(\square) = 3 \end{cases}$$



At each vertex $\square$, 2 possible configurations :

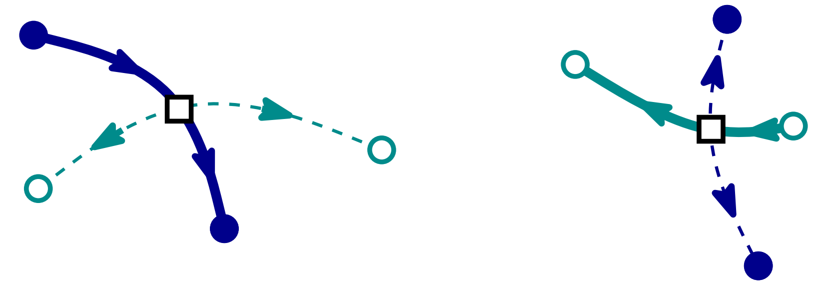


$|\{\text{blue edges}\}| = |V(M)| - 1$

To prove that it is a tree, enough to prove that there is no cycles (exercise !).

Lattice structure on the set of spanning trees

$$\begin{cases} \alpha_T(\circledcirc) = \alpha_T(\odot) = 0 \\ \alpha_T(\circ) = \alpha_T(\bullet) = 1 \\ \alpha_T(\square) = 3 \end{cases}$$

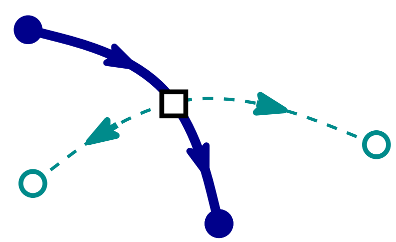


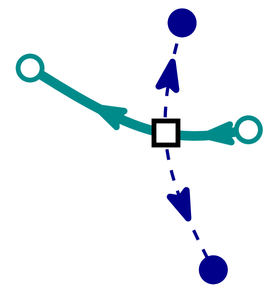
Which cycles are essential ?

Which edges are rigid ?

Lattice structure on the set of spanning trees

$$\begin{cases} \alpha_T(\circledcirc) = \alpha_T(\odot) = 0 \\ \alpha_T(\circ) = \alpha_T(\bullet) = 1 \\ \alpha_T(\square) = 3 \end{cases}$$





Which cycles are essential ?

Which edges are rigid ?

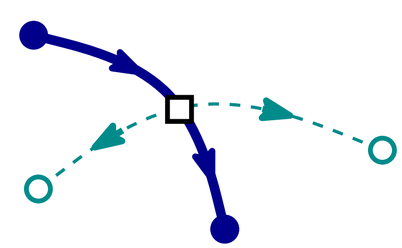
To make our life easier, assume that M is bridgeless.

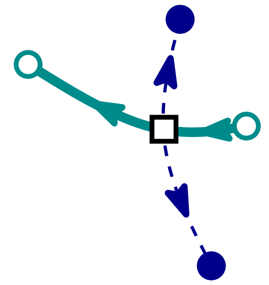
Property :

The rigid edges are the edges incident to the root vertex.

Lattice structure on the set of spanning trees

$$\begin{cases} \alpha_T(\circledcirc) = \alpha_T(\odot) = 0 \\ \alpha_T(\circ) = \alpha_T(\bullet) = 1 \\ \alpha_T(\square) = 3 \end{cases}$$



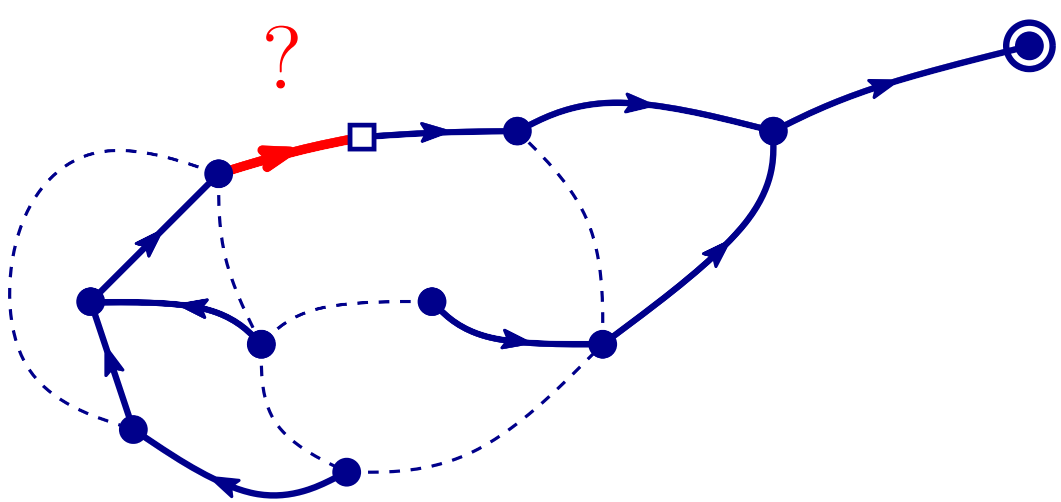


Which cycles are essential ? Which edges are rigid ?

To make our life easier, assume that M is bridgeless.

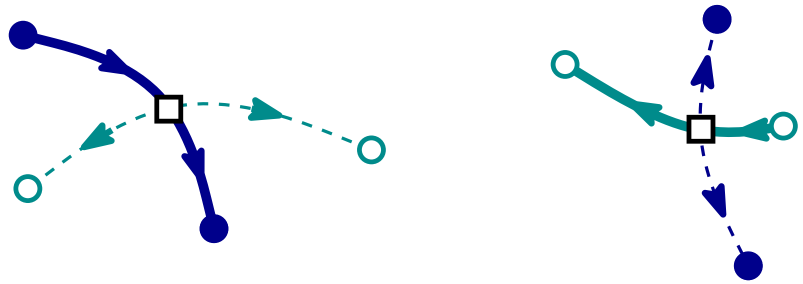
Property :

The rigid edges are the edges incident to the root vertex.



Lattice structure on the set of spanning trees

$$\begin{cases} \alpha_T(\circledcirc) = \alpha_T(\odot) = 0 \\ \alpha_T(\circ) = \alpha_T(\bullet) = 1 \\ \alpha_T(\square) = 3 \end{cases}$$

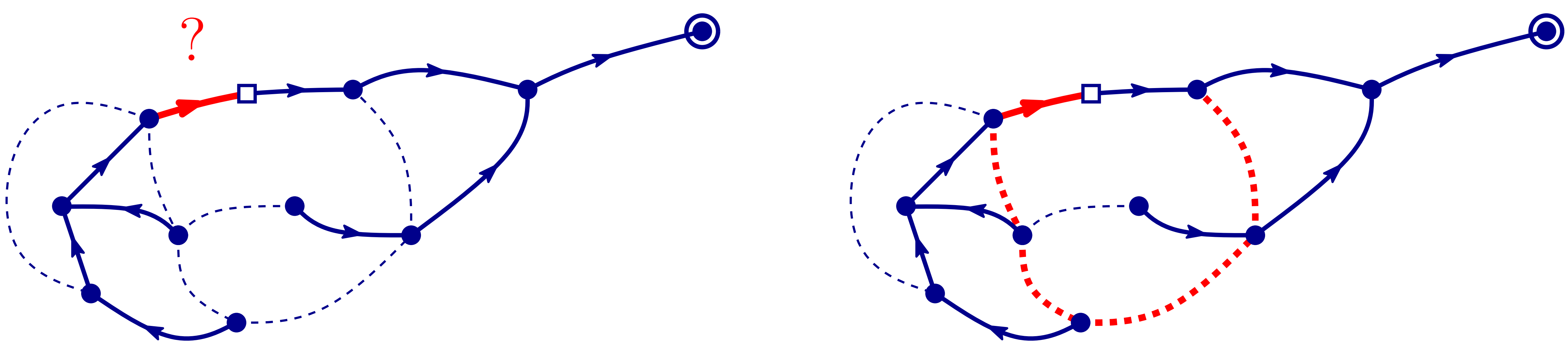


Which cycles are essential ? Which edges are rigid ?

To make our life easier, assume that M is bridgeless.

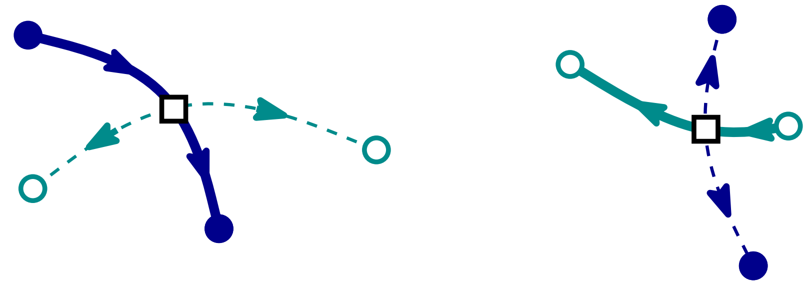
Property :

The rigid edges are the edges incident to the root vertex.



Lattice structure on the set of spanning trees

$$\begin{cases} \alpha_T(◎) = \alpha_T(⦿) = 0 \\ \alpha_T(○) = \alpha_T(●) = 1 \\ \alpha_T(□) = 3 \end{cases}$$

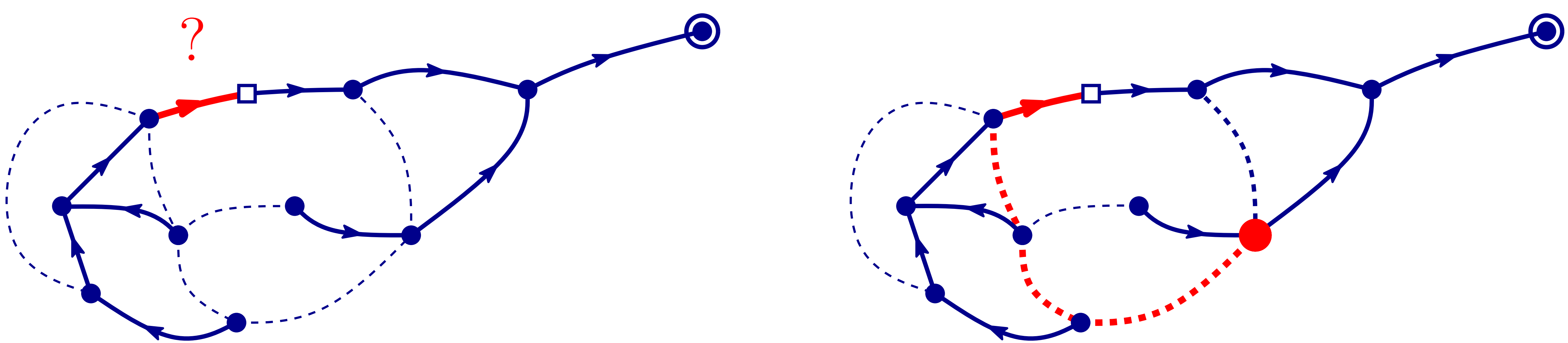


Which cycles are essential ? Which edges are rigid ?

To make our life easier, assume that M is bridgeless.

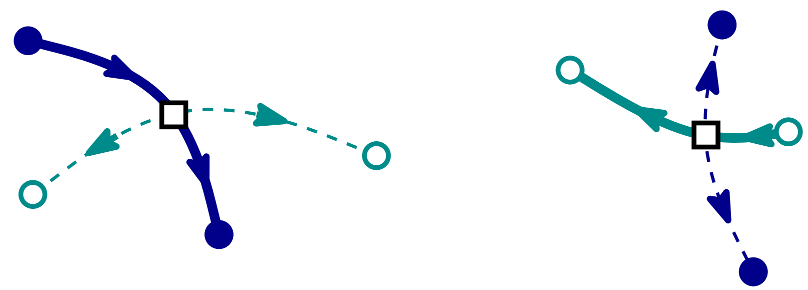
Property :

The rigid edges are the edges incident to the root vertex.



Lattice structure on the set of spanning trees

$$\begin{cases} \alpha_T(\circledcirc) = \alpha_T(\odot) = 0 \\ \alpha_T(\circ) = \alpha_T(\bullet) = 1 \\ \alpha_T(\square) = 3 \end{cases}$$



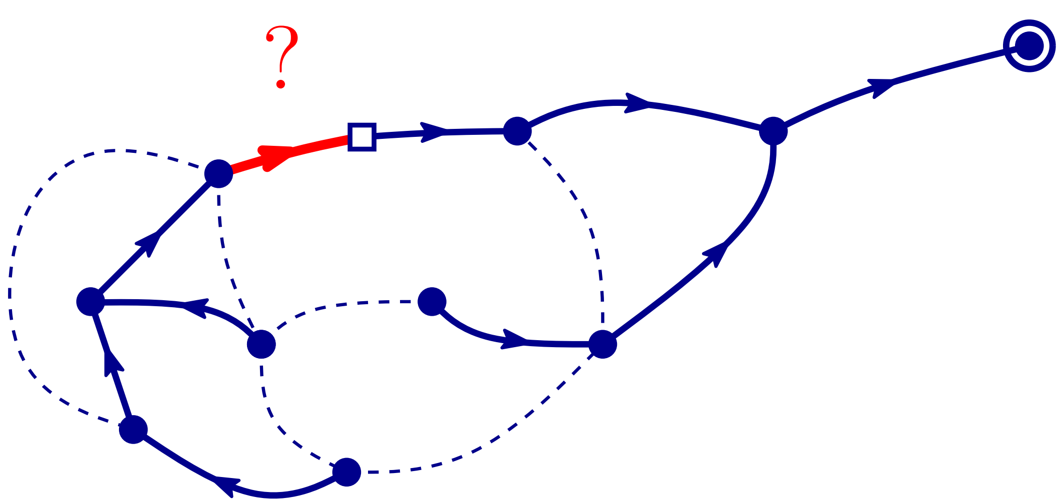
Which cycles are essential ?

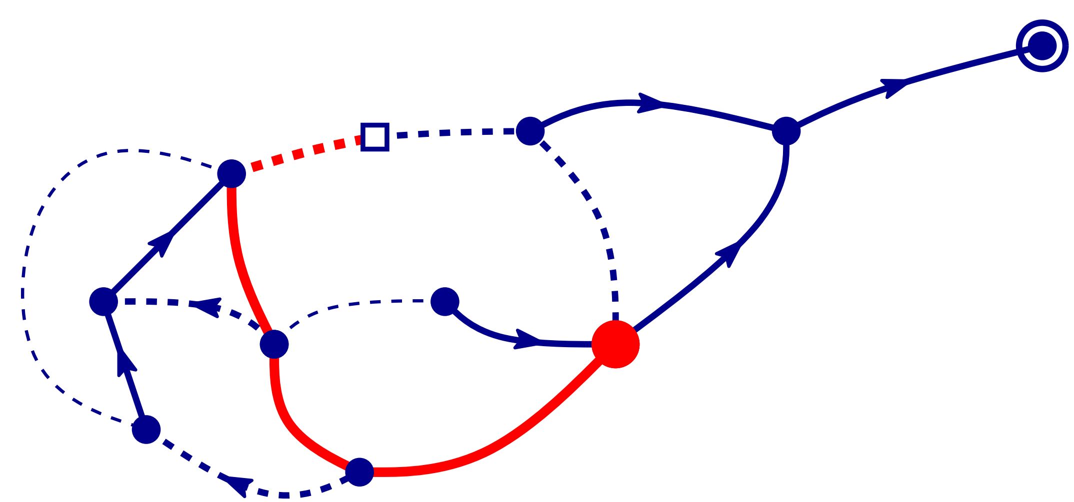
Which edges are rigid ?

To make our life easier, assume that M is bridgeless.

Property :

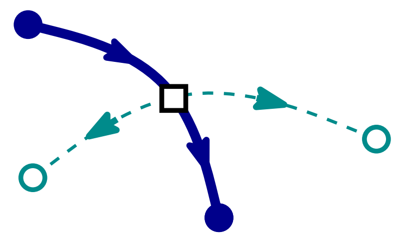
The rigid edges are the edges incident to the root vertex.

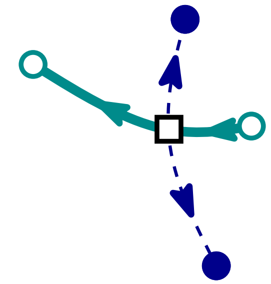




Lattice structure on the set of spanning trees

$$\begin{cases} \alpha_T(\circledcirc) = \alpha_T(\odot) = 0 \\ \alpha_T(\circ) = \alpha_T(\bullet) = 1 \\ \alpha_T(\square) = 3 \end{cases}$$





Which cycles are essential ? Which edges are rigid ?

To make our life easier, assume that M is bridgeless.

Property :

The rigid edges are the edges incident to the root vertex.

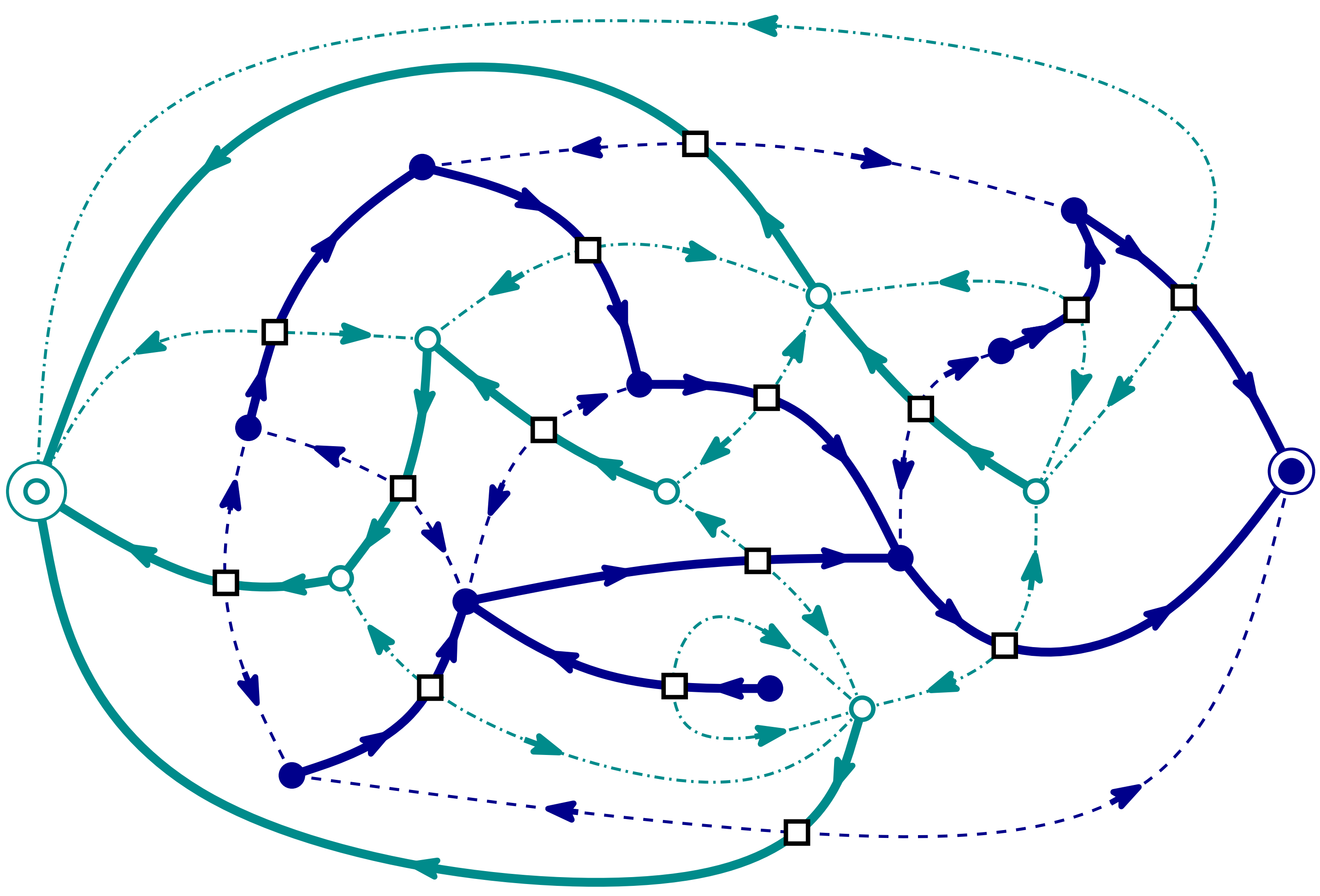
Property :

The essential cycles are the facial cycles of $\tilde{M}$, without root vertices.

Lattice structure on the set of spanning trees

Property :

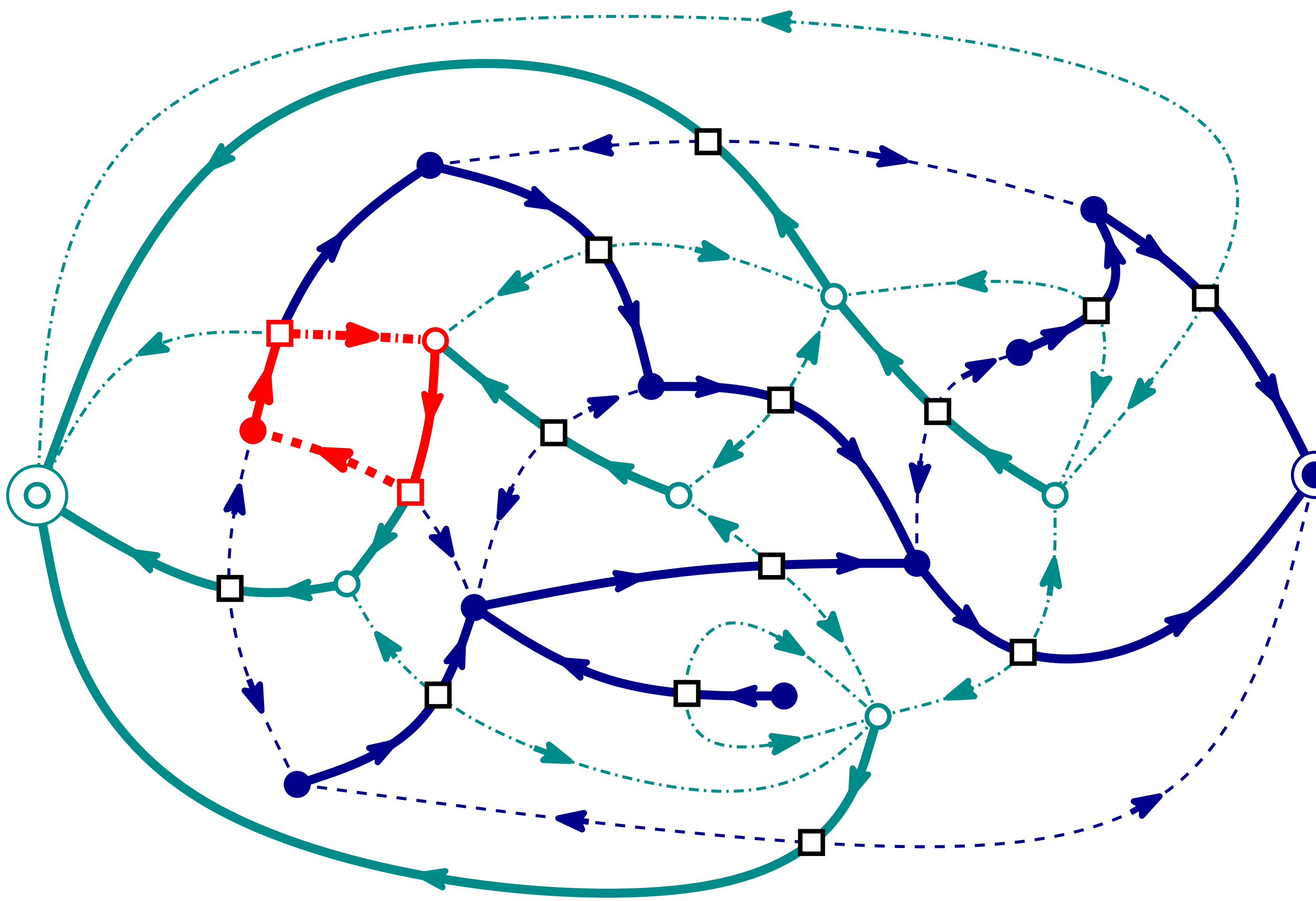
The essential cycles are the facial cycles of $\tilde{M}$, without root vertices.

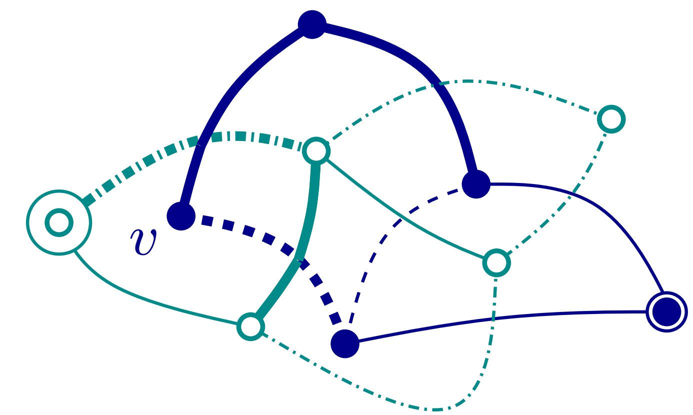


Lattice structure on the set of spanning trees

Property :

The essential cycles are the facial cycles of $\tilde{M}$, without root vertices.

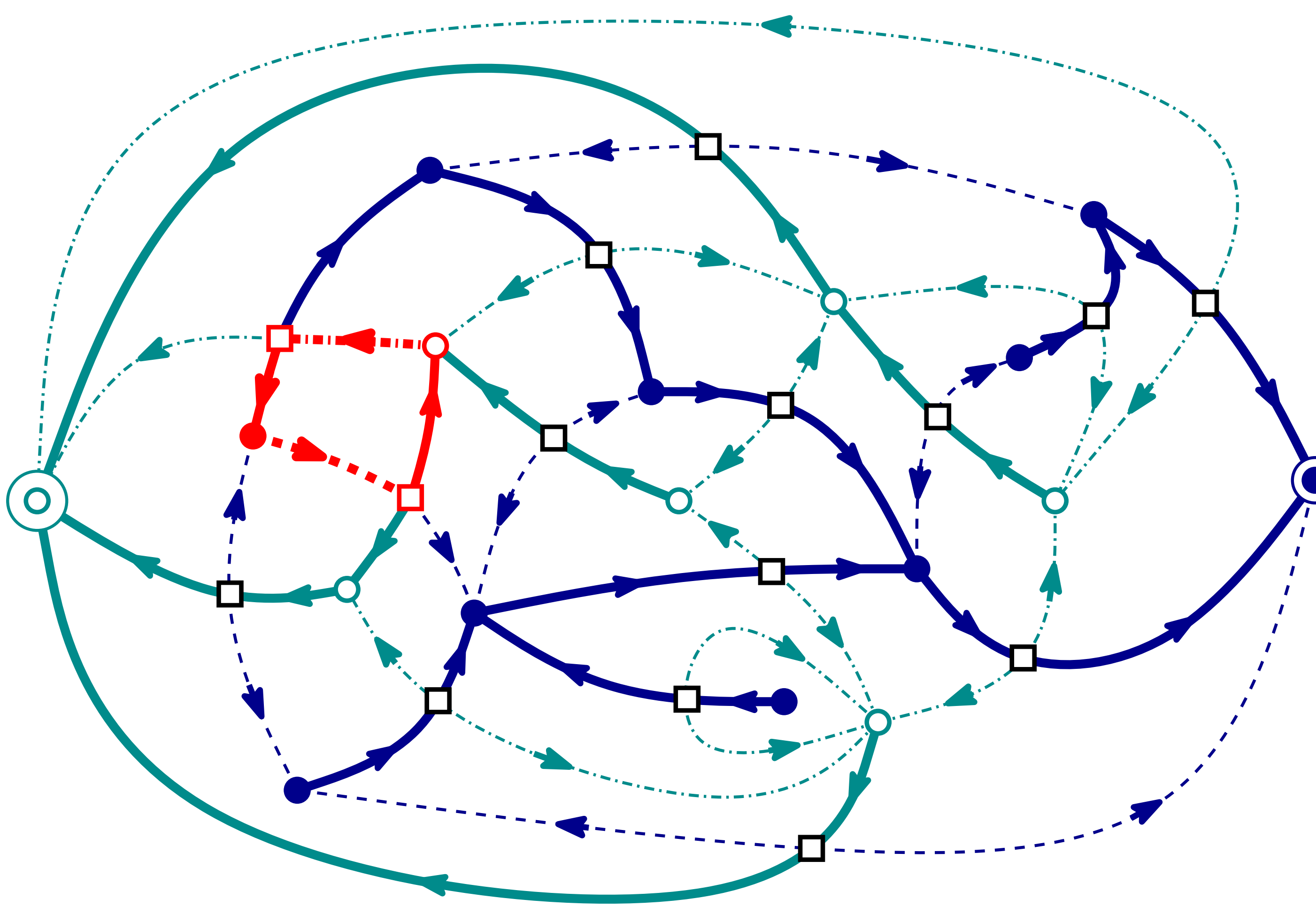


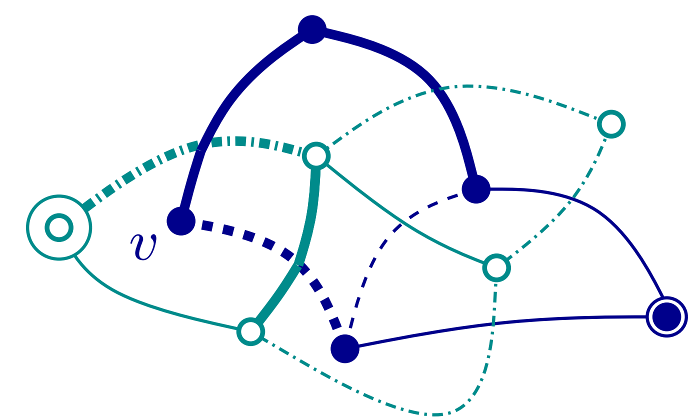


Lattice structure on the set of spanning trees

Property :

The essential cycles are the facial cycles of $\tilde{M}$, without root vertices.

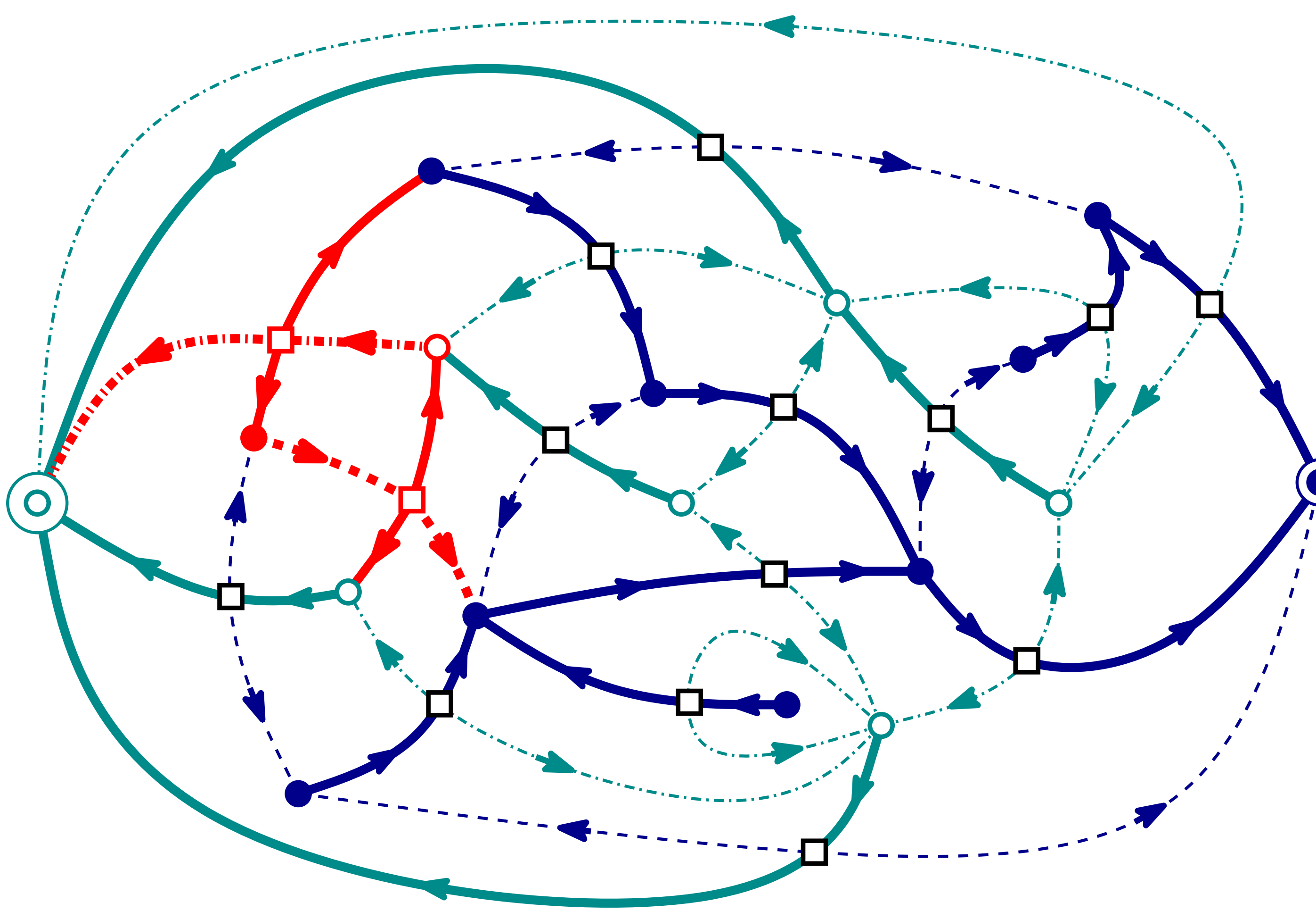


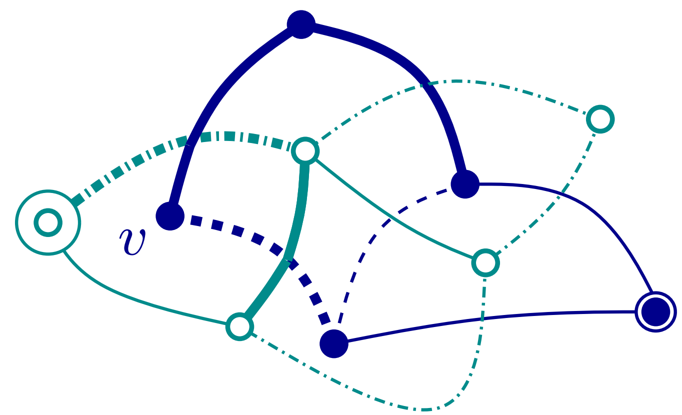


Lattice structure on the set of spanning trees

Property :

The essential cycles are the facial cycles of $\tilde{M}$, without root vertices.

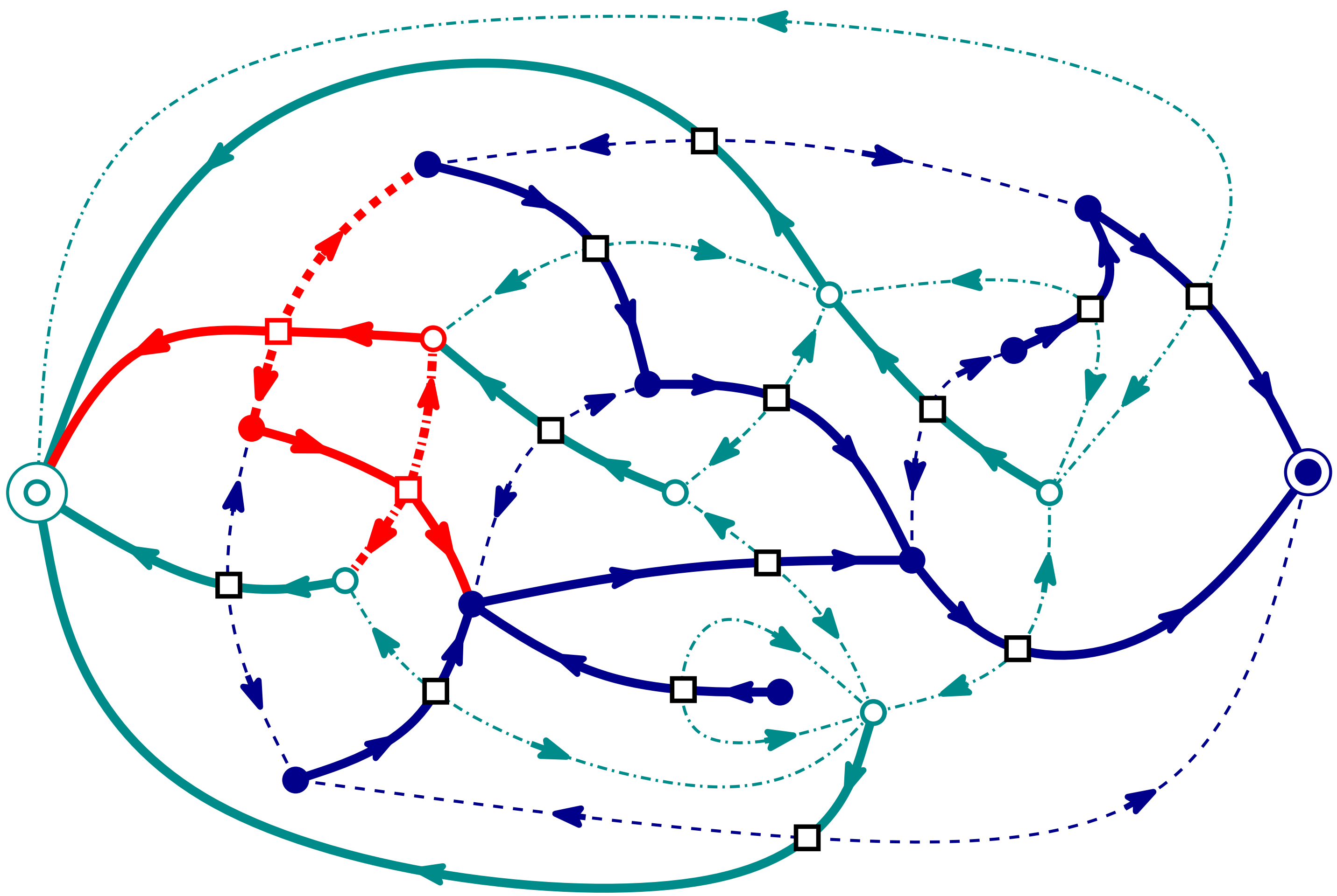


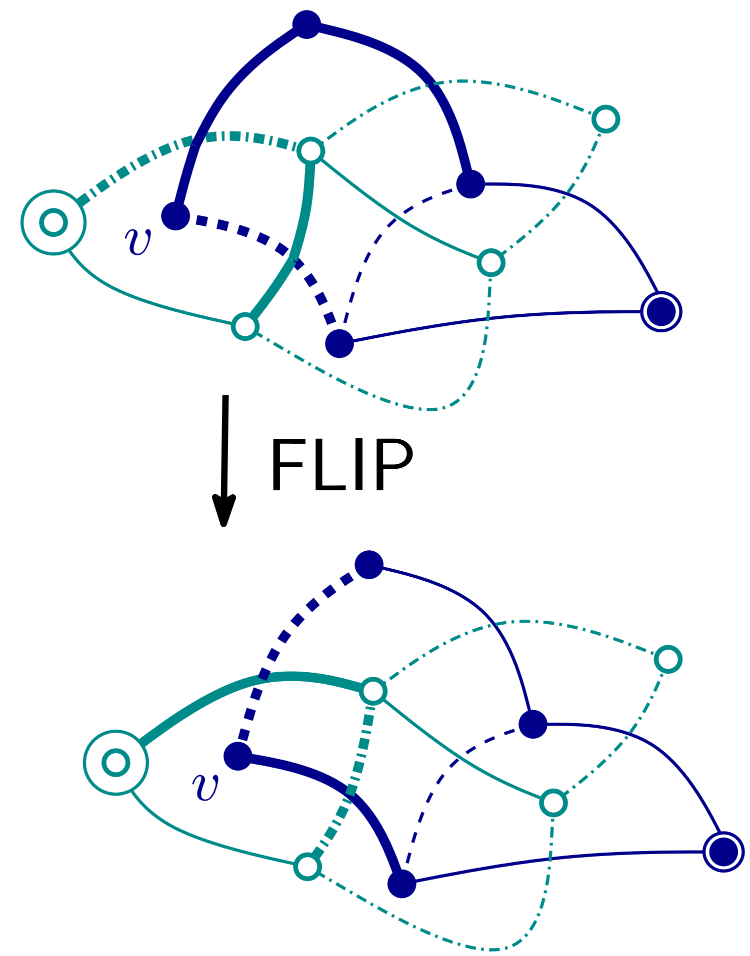


Lattice structure on the set of spanning trees

Property :

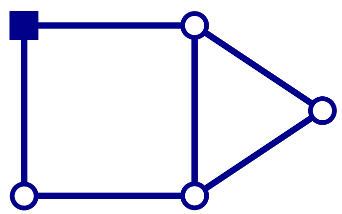
The essential cycles are the facial cycles of $\tilde{M}$, without root vertices.

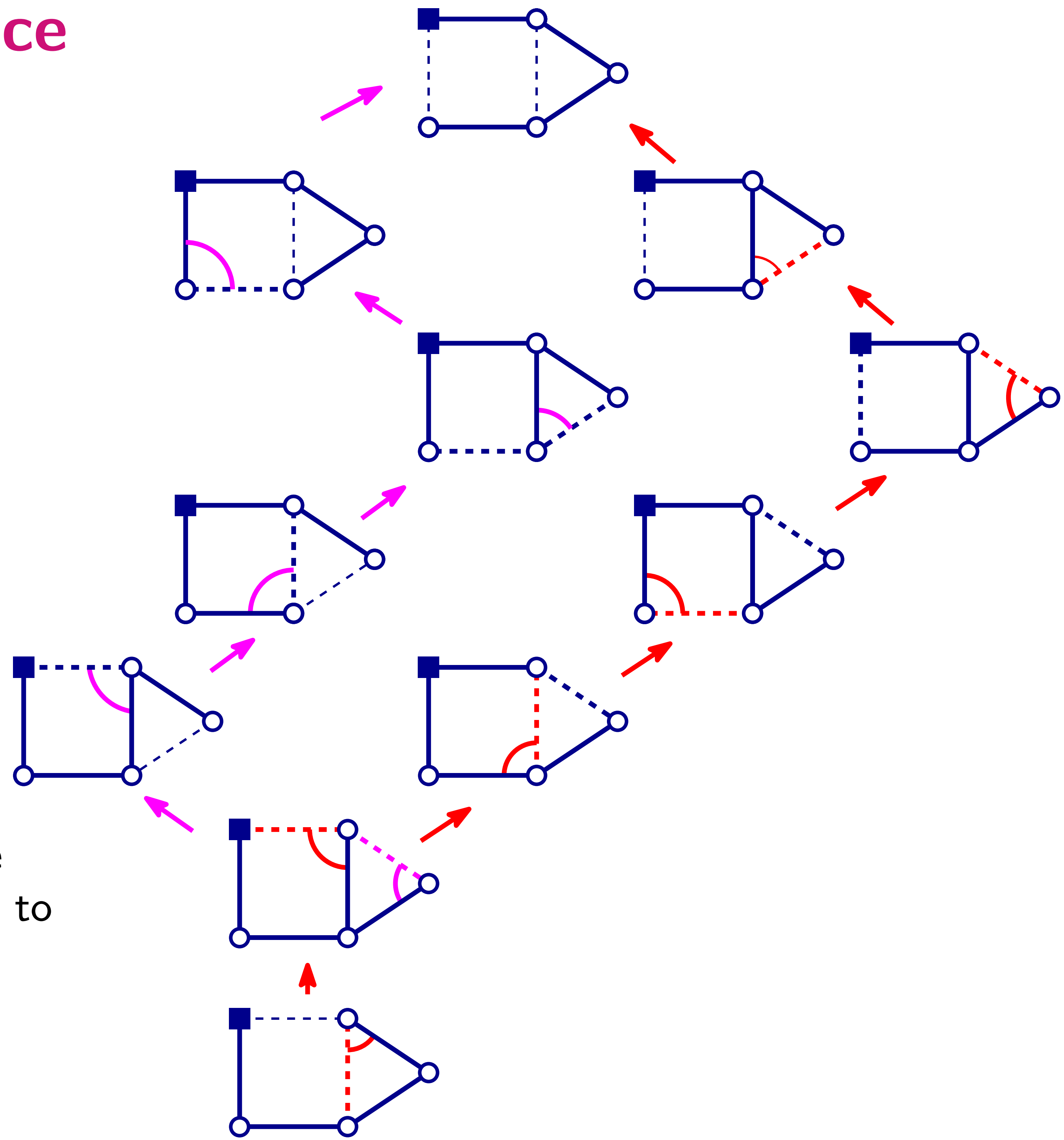




Example of a lattice

M



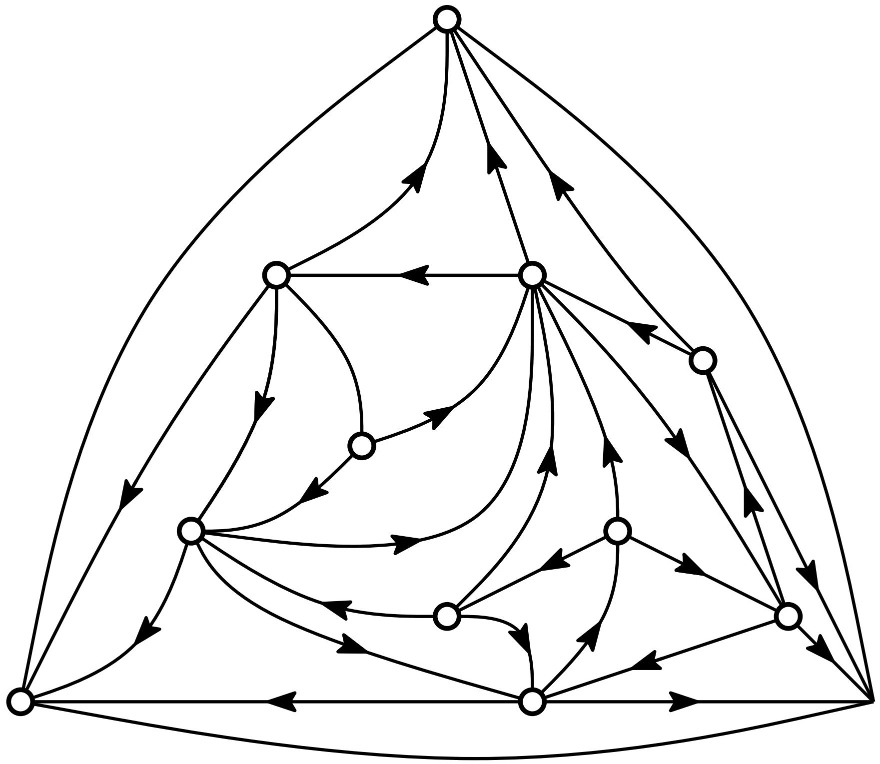


Applications :

Coupling from the past for distributive lattices.

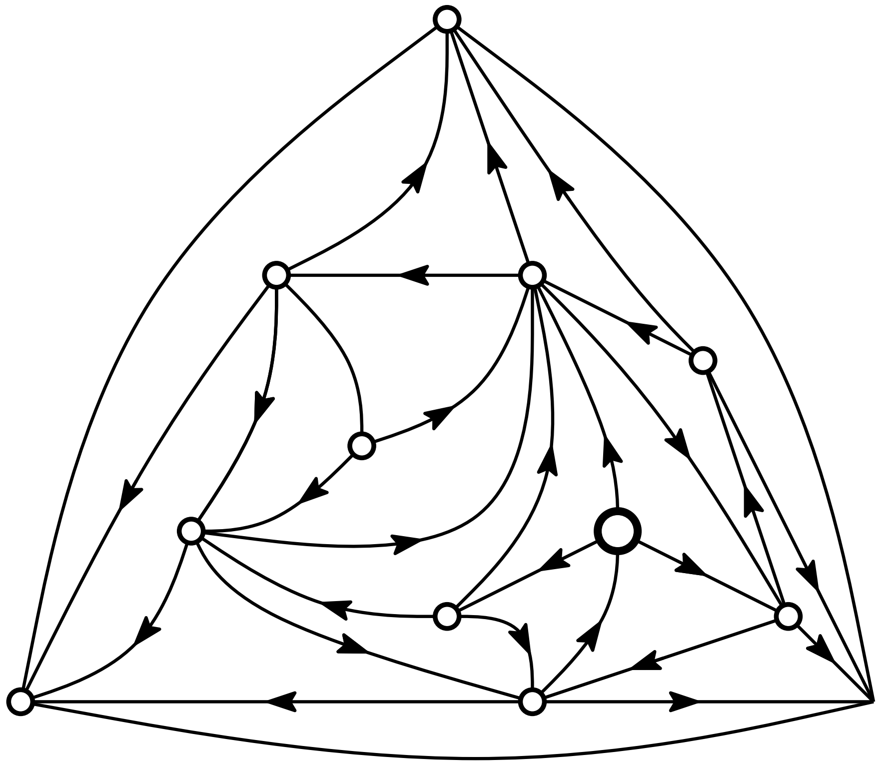
''easy'' to sample a spanning tree (Wilson's algorithm), gives a way to sample some perfect matchings (a.k.a. dimer models) [Kenyon, Propp, Wilson]

Schnyder woods



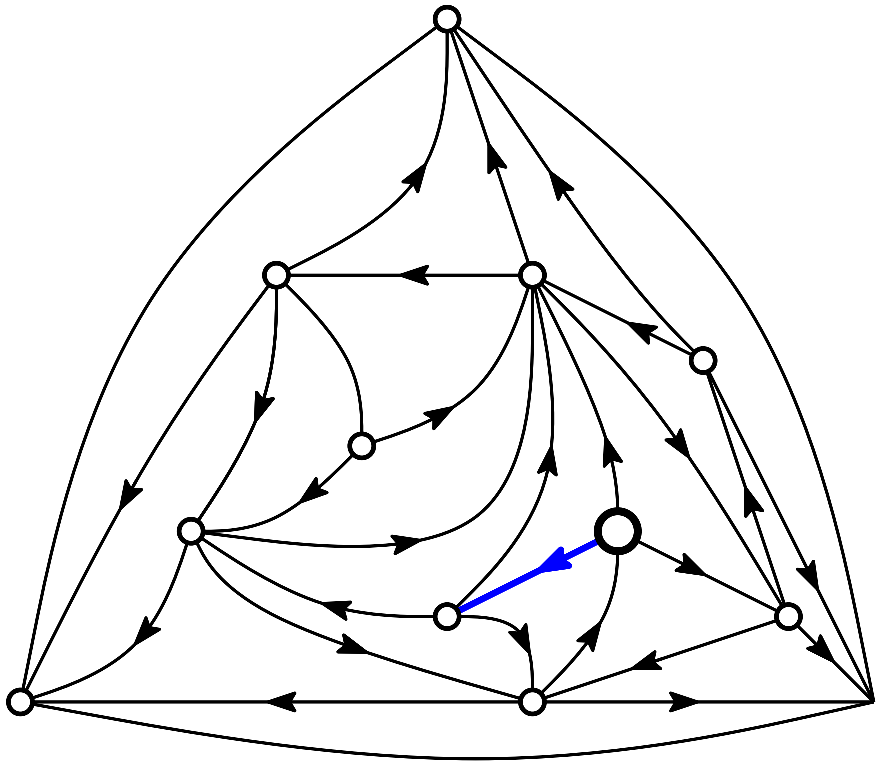
Simple triangulation endowed with a 3-orientation.

Schnyder woods



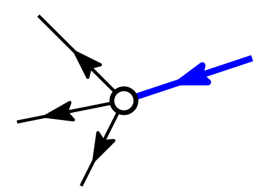
Simple triangulation endowed with a 3-orientation.

Schnyder woods

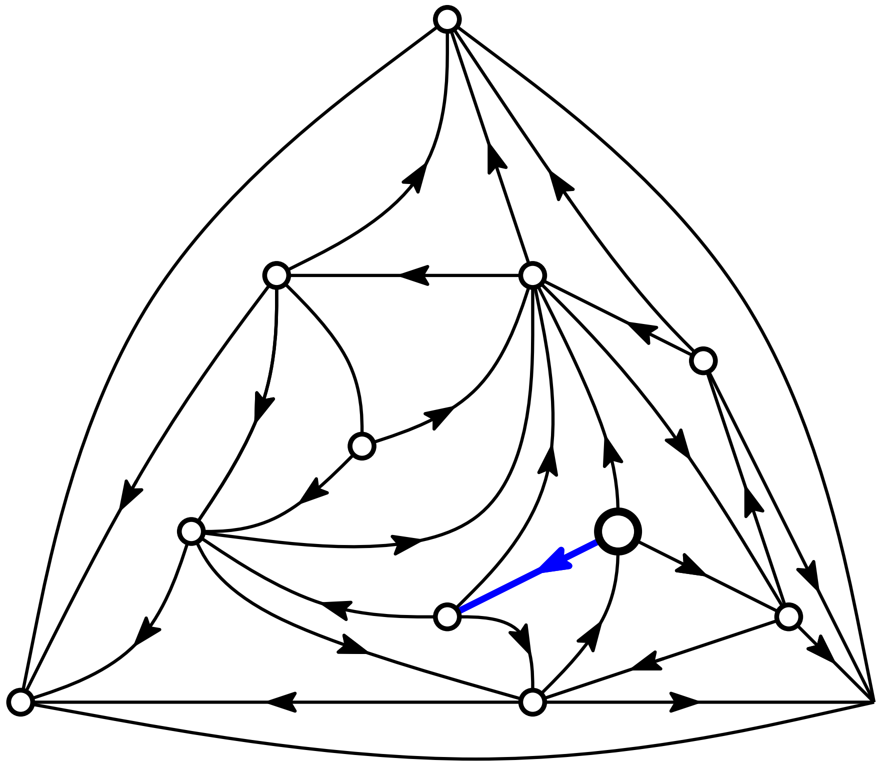


Simple triangulation endowed with a 3-orientation.

Consider the ‘’middle”-paths

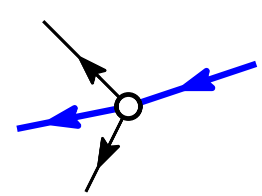


Schnyder woods

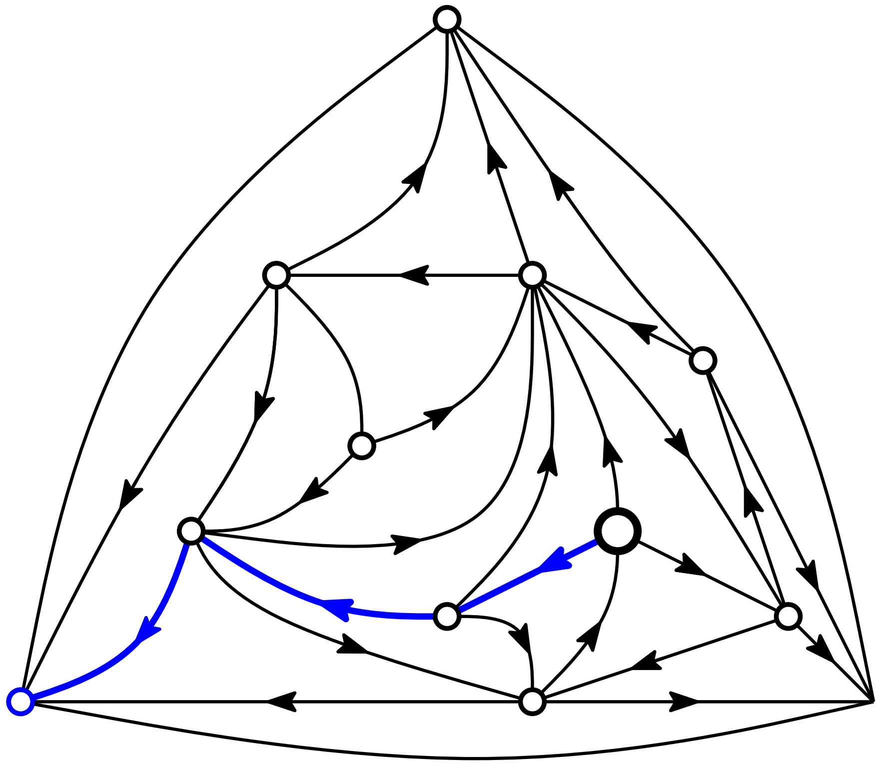


Simple triangulation endowed with a 3-orientation.

Consider the ''middle''-paths

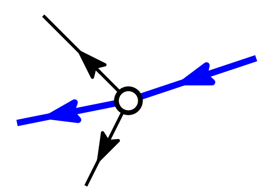


Schnyder woods

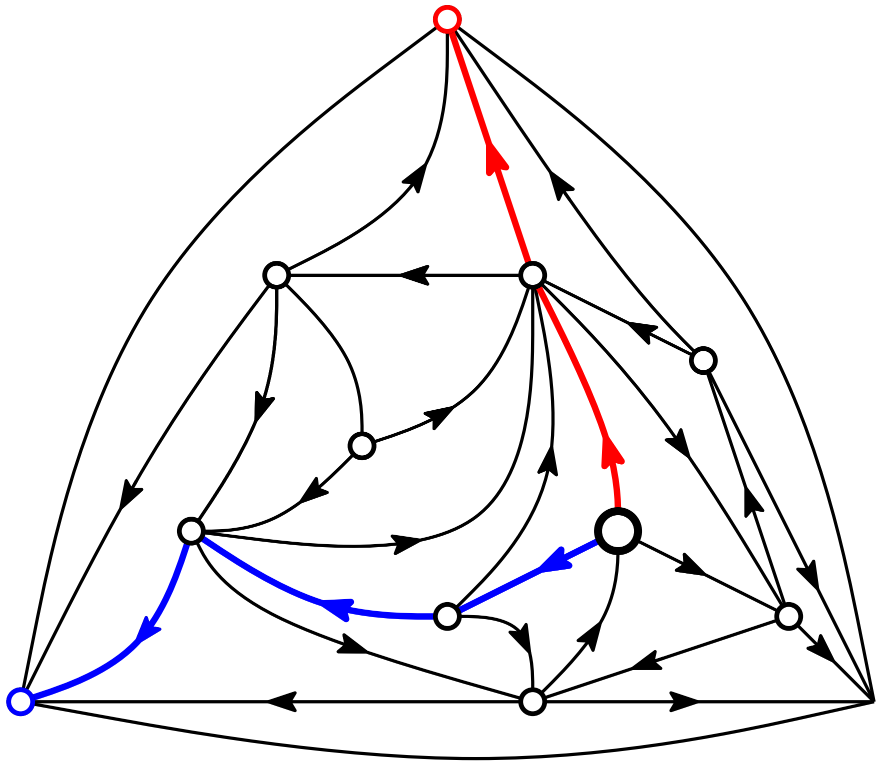


Simple triangulation endowed with a 3-orientation.

Consider the ''middle''-paths

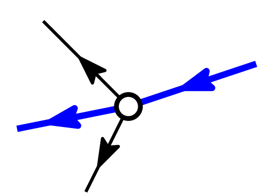


Schnyder woods

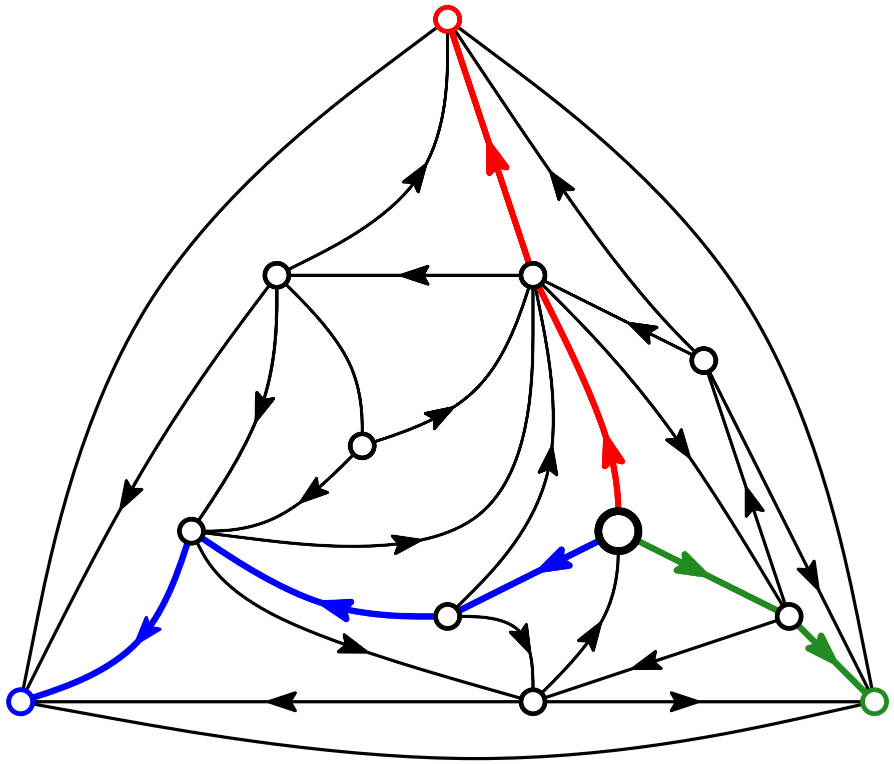


Simple triangulation endowed with a 3-orientation.

Consider the ‘’middle”-paths

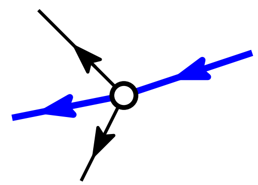


Schnyder woods



Simple triangulation endowed with a 3-orientation.

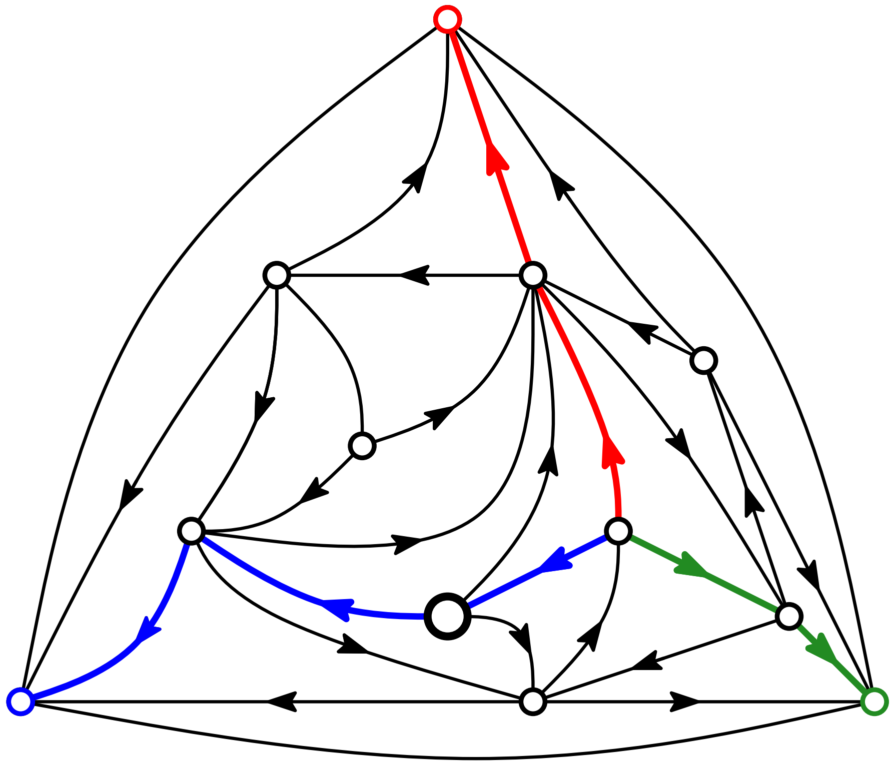
Consider the ''middle''-paths



Property :

The middle-paths are self-avoiding + the 3 middle paths starting at a given vertex do not intersect one another.

Schnyder woods

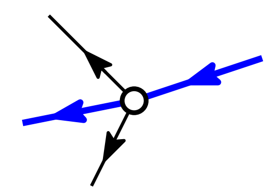


Simple triangulation endowed with a 3-orientation.

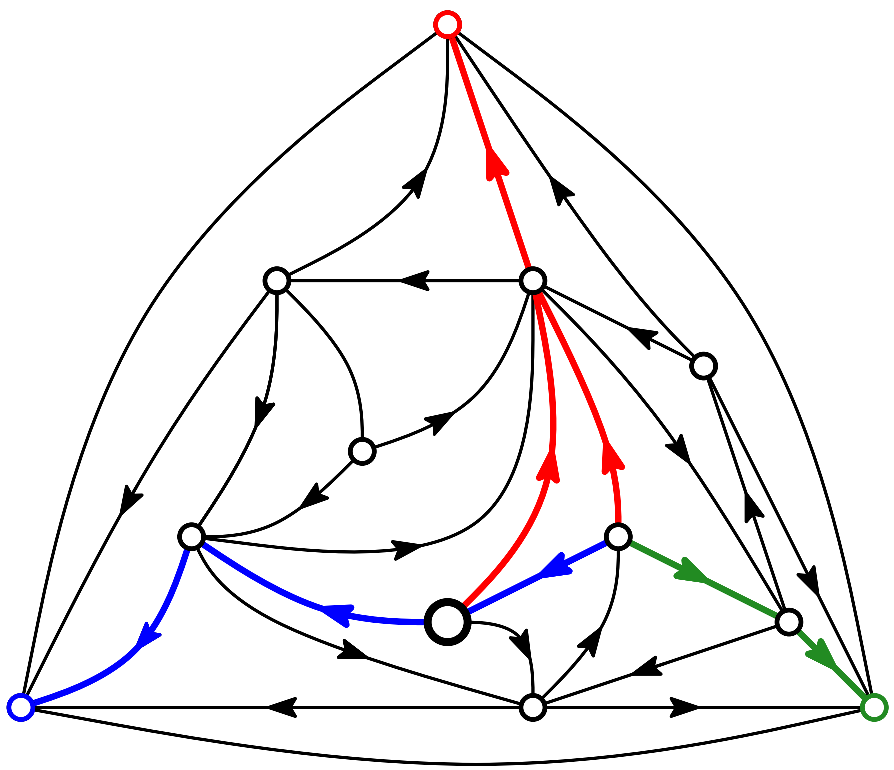
Consider the ''middle''-paths

Property :

The middle-paths are self-avoiding + the 3 middle paths starting at a given vertex do not intersect one another.

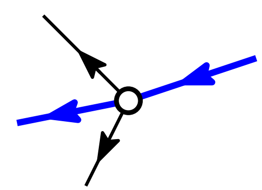


Schnyder woods



Simple triangulation endowed with a 3-orientation.

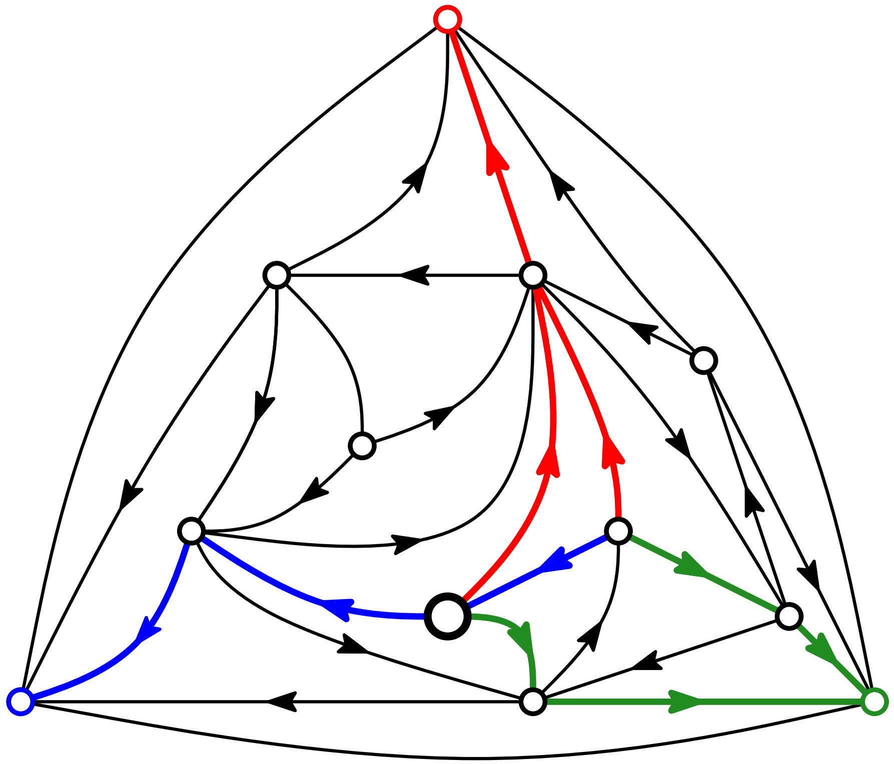
Consider the ''middle''-paths



Property :

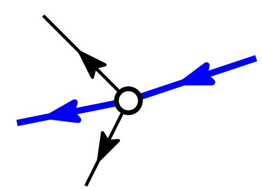
The middle-paths are self-avoiding + the 3 middle paths starting at a given vertex do not intersect one another.

Schnyder woods



Simple triangulation endowed with a 3-orientation.

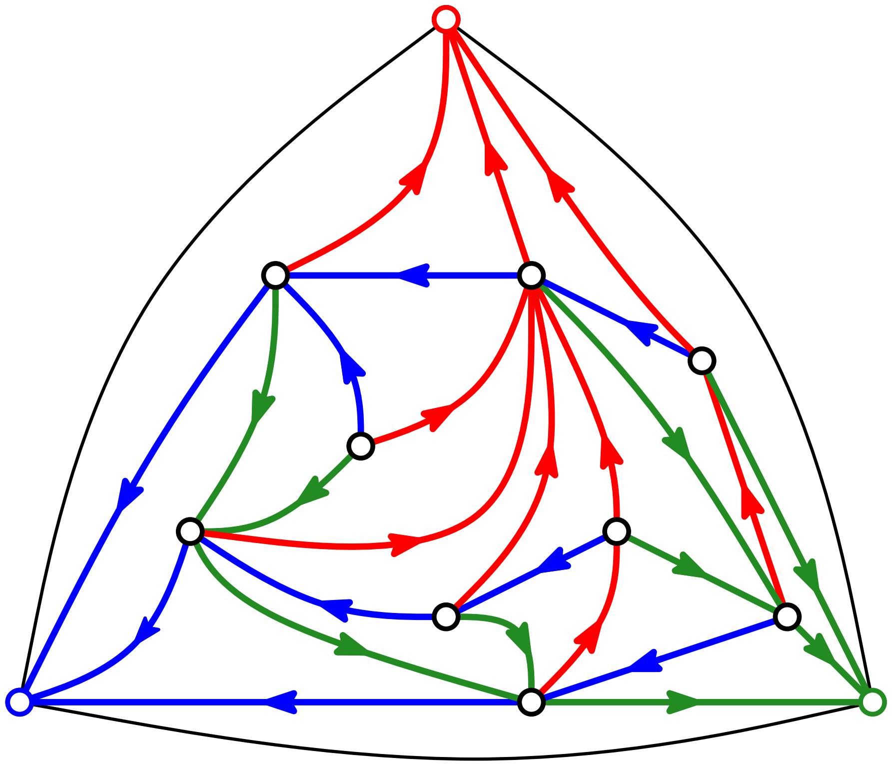
Consider the ‘’middle”-paths



Property :

The middle-paths are self-avoiding + the 3 middle paths starting at a given vertex do not intersect one another.

Schnyder woods

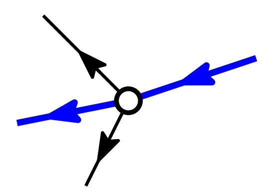


Simple triangulation endowed with a 3-orientation.

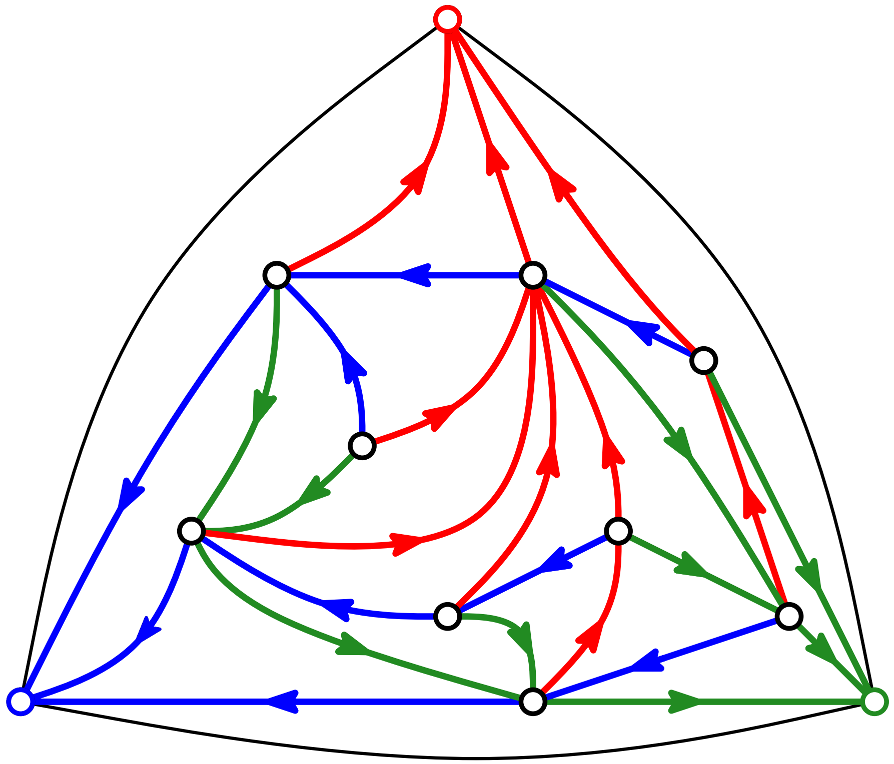
Consider the ''middle''-paths

Property :

The middle-paths are self-avoiding + the 3 middle paths starting at a given vertex do not intersect one another.



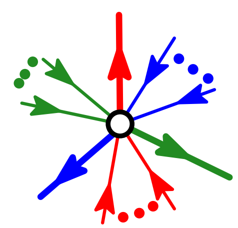
Schnyder woods



Simple triangulation endowed with a 3-orientation.

Consider the ''middle''-paths

Around each inner vertex :

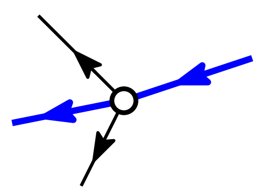


Property :

The middle-paths are self-avoiding + the 3 middle paths starting at a given vertex do not intersect one another.

Property :

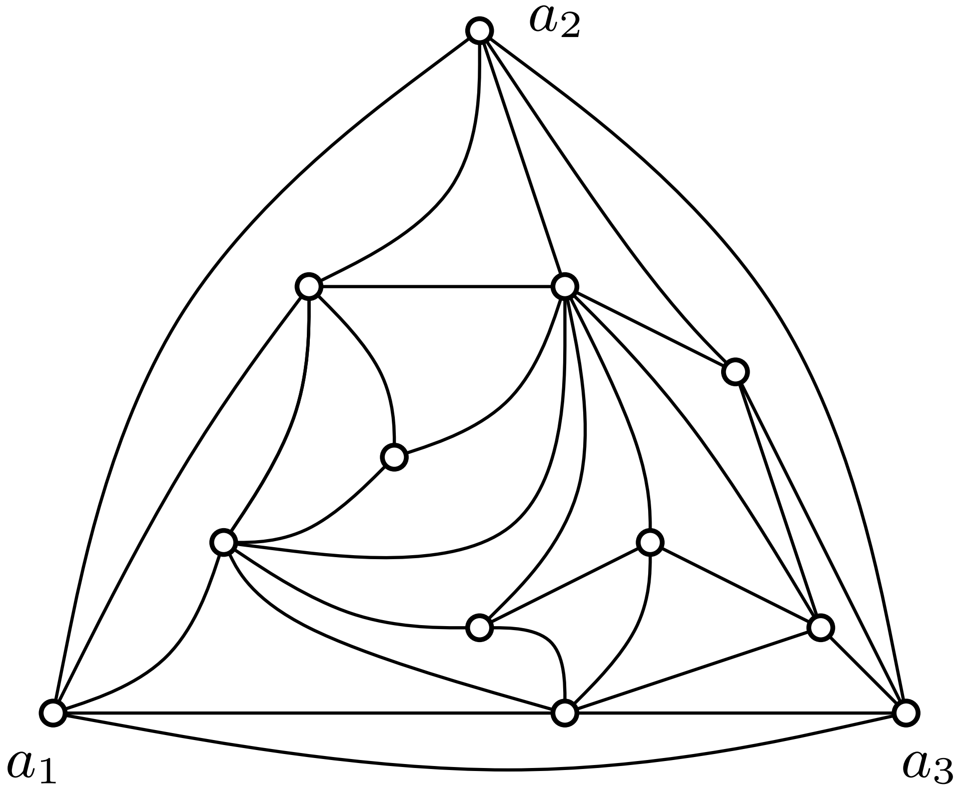
The subset of edges of a given color is a spanning tree of the inner vertices (+ one outer vertex) of the triangulation.



This coloring is a **Schnyder wood**.

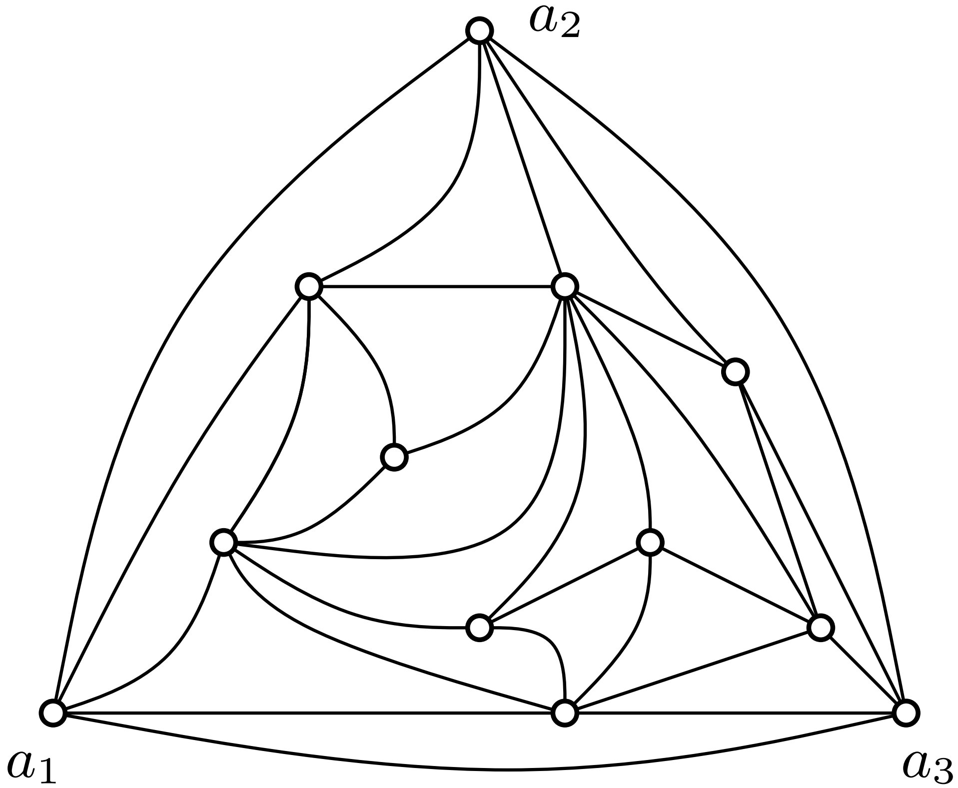
[Schnyder '89]

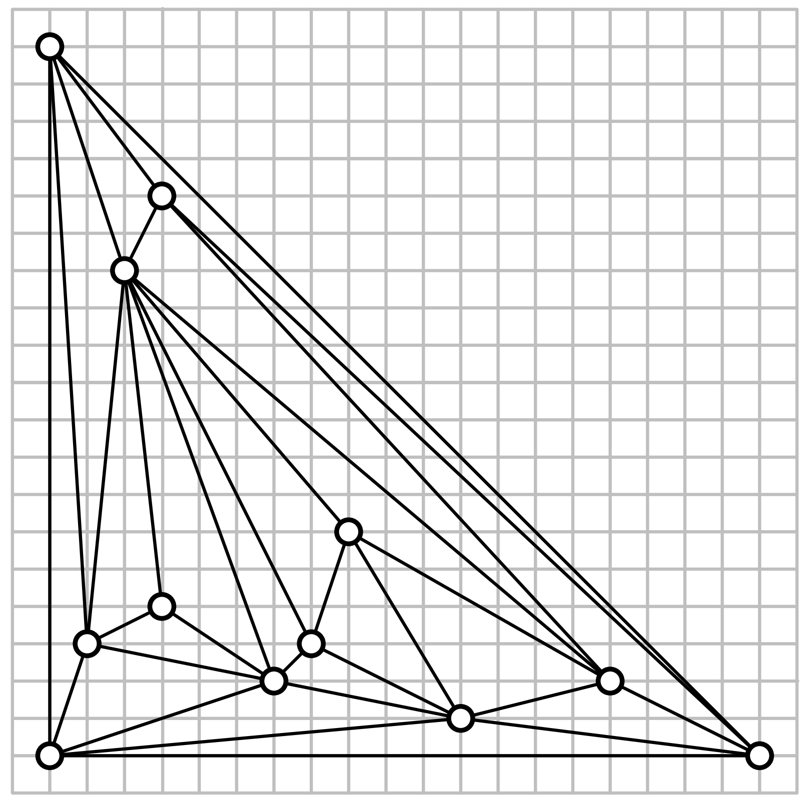
Application to straight-line drawing.



Goal : Find a representation of the plane map in which all the edges are straight lines and where the vertices lie on a grid.

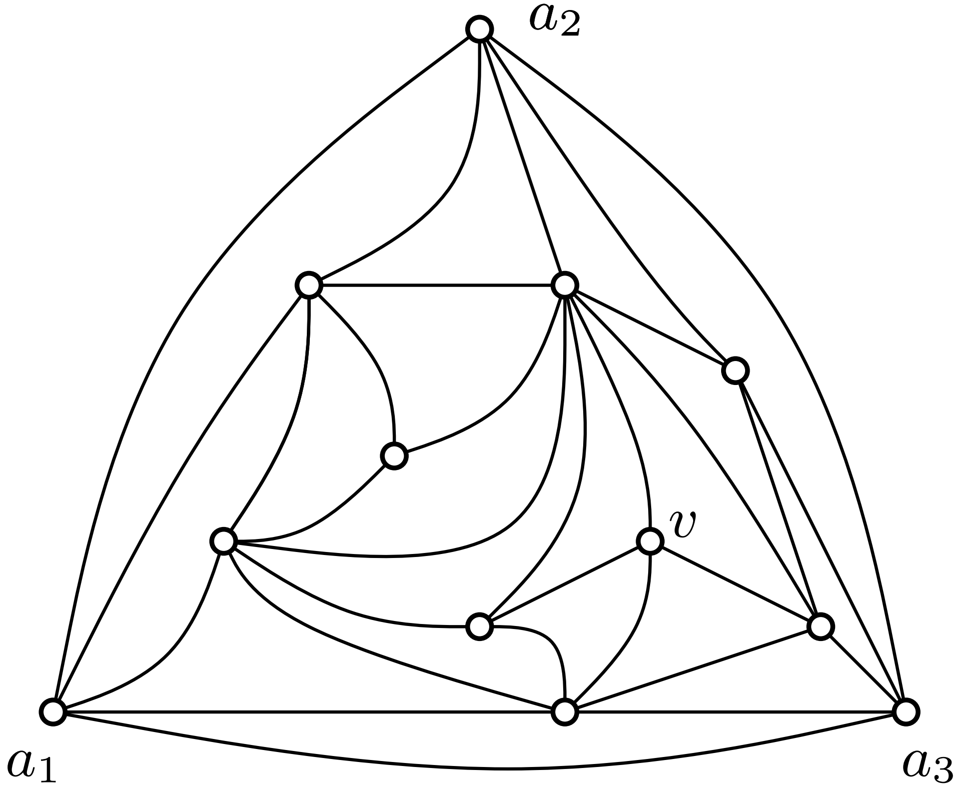
Application to straight-line drawing.

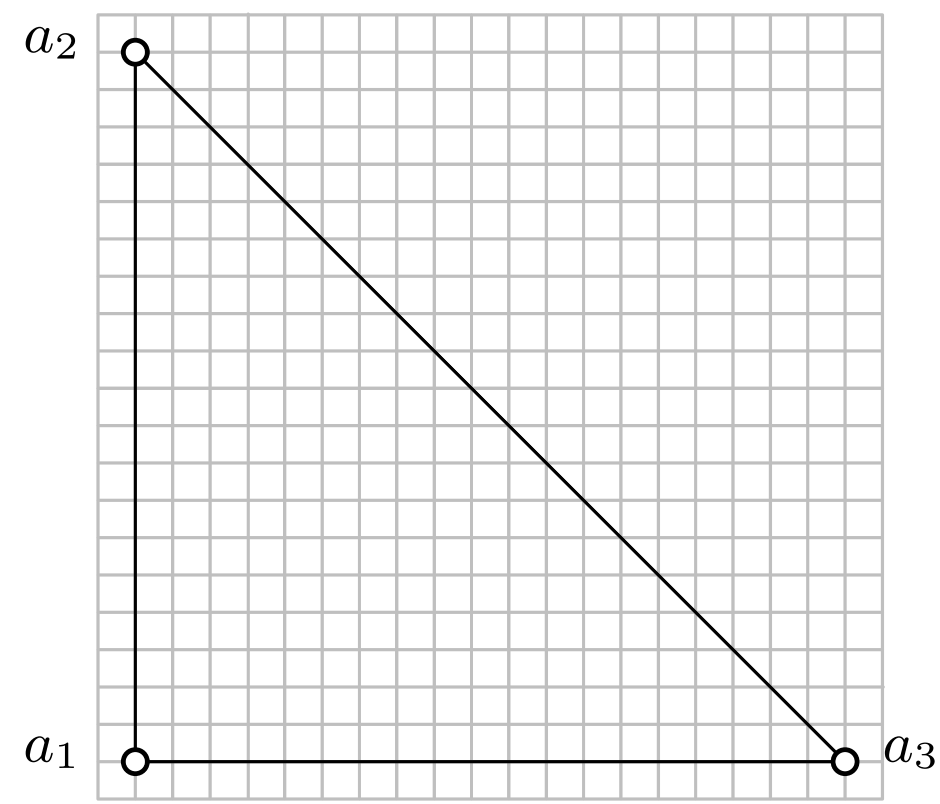




Goal : Find a representation of the plane map in which all the edges are straight lines and where the vertices lie on a grid.

Application to straight-line drawing.

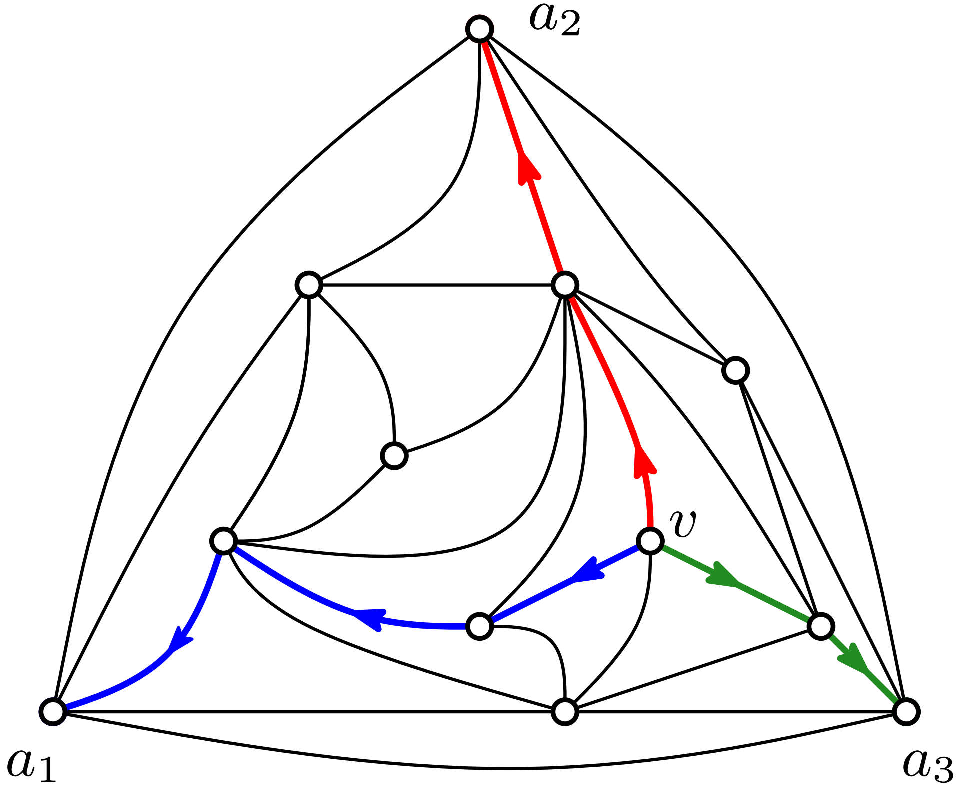


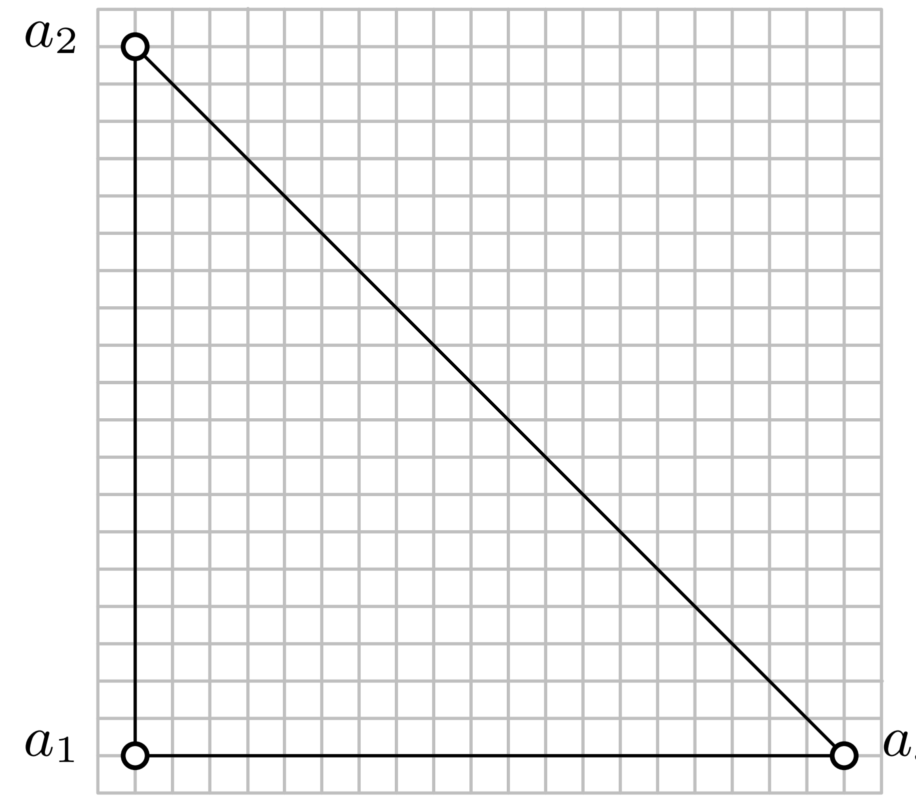


Goal : Find a representation of the plane map in which all the edges are straight lines and where the vertices lie on a grid.

Algo :
- Put the vertices a_1, a_2 et a_3 at positions $(0,0)$, $(0,|F(M)|)$, $(|F(M)|,0)$.

Application to straight-line drawing.



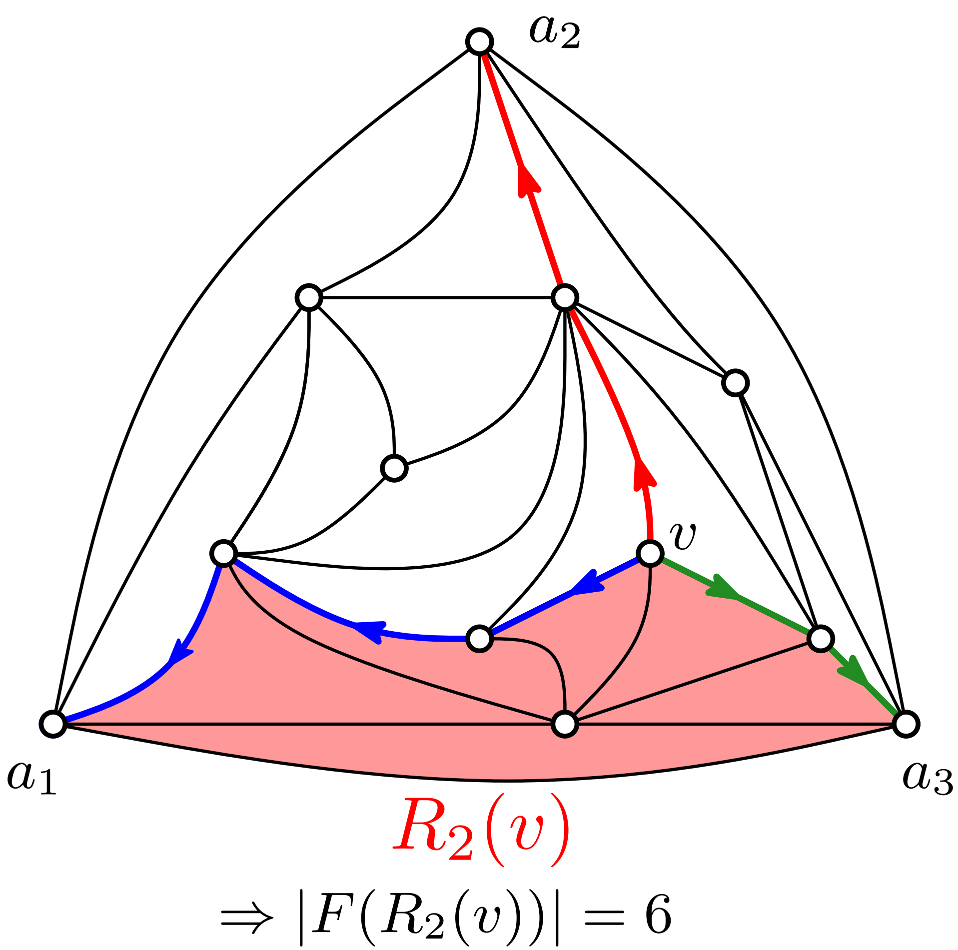


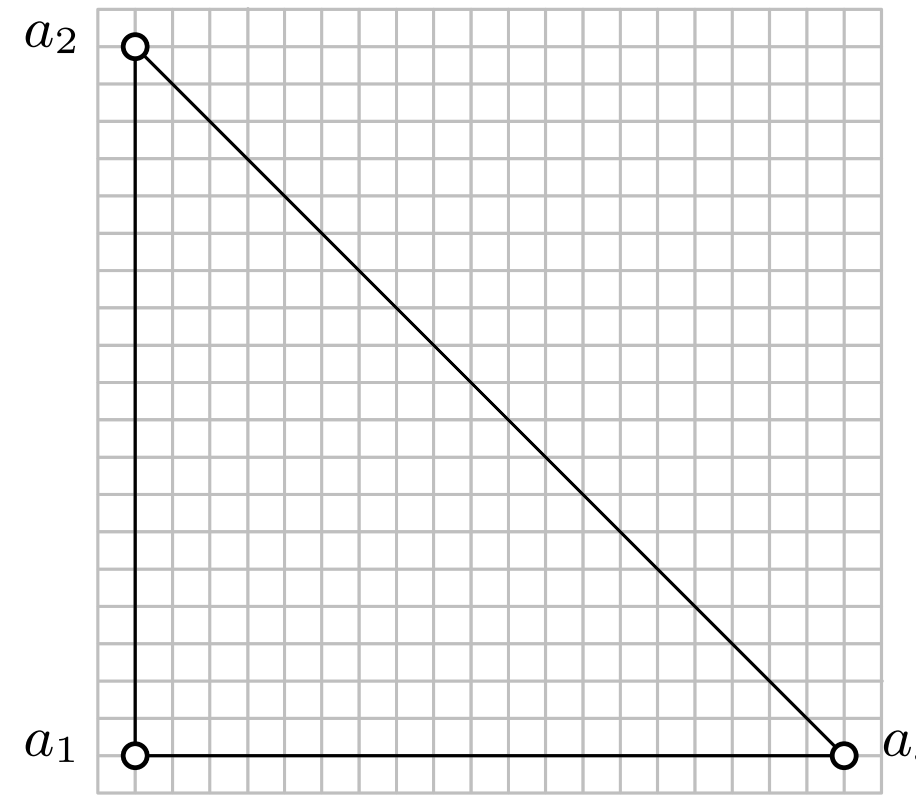
Goal : Find a representation of the plane map in which all the edges are straight lines and where the vertices lie on a grid.

Algo :

- Put the vertices a_1, a_2 et a_3 at positions $(0,0)$, $(0,|F(M)|)$, $(|F(M)|,0)$.
- Put each inner vertex v at position $(|F(R_3(v))|, |F(R_2(v))|)$.

Application to straight-line drawing.

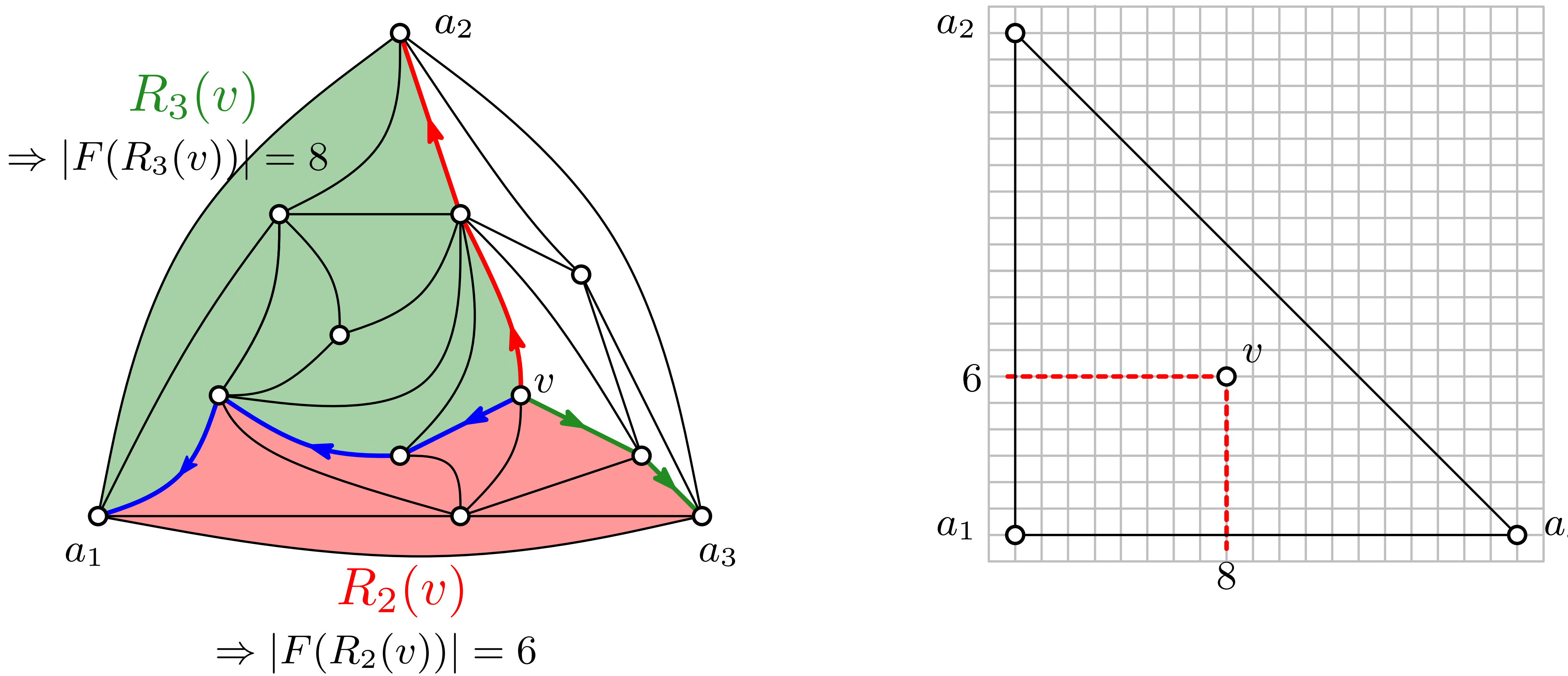




Algo :

- Put the vertices a_1, a_2 et a_3 at positions $(0,0)$, $(0,|F(M)|)$, $(|F(M)|,0)$.
- Put each inner vertex v at position $(|F(R_3(v))|, |F(R_2(v))|)$.

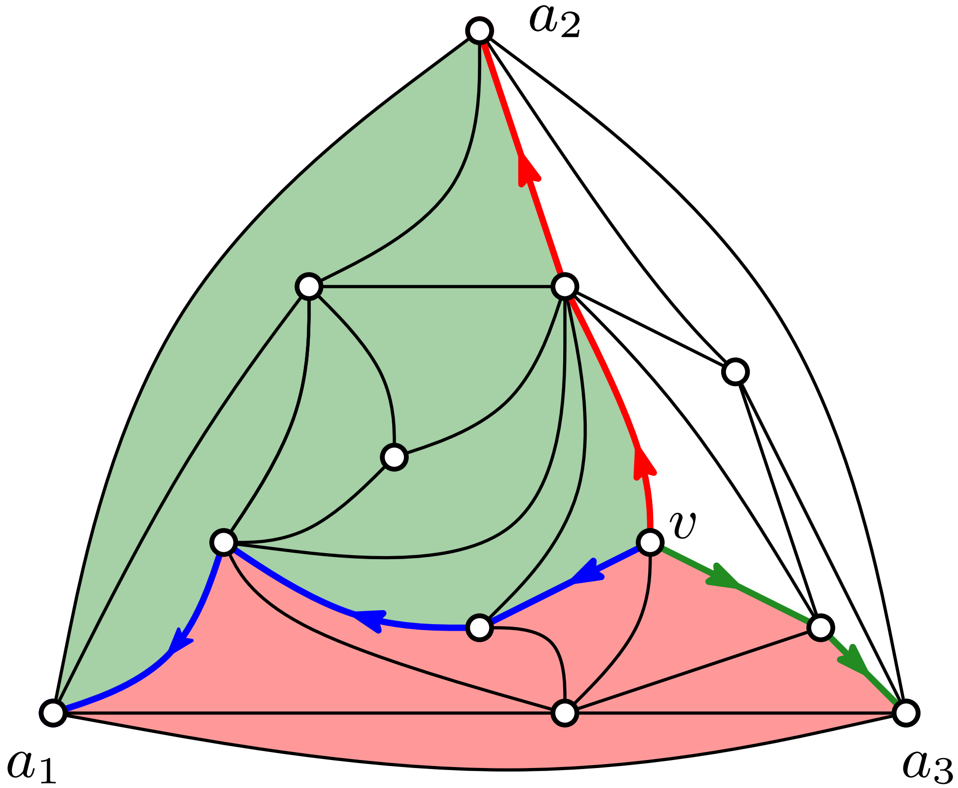
Application to straight-line drawing.

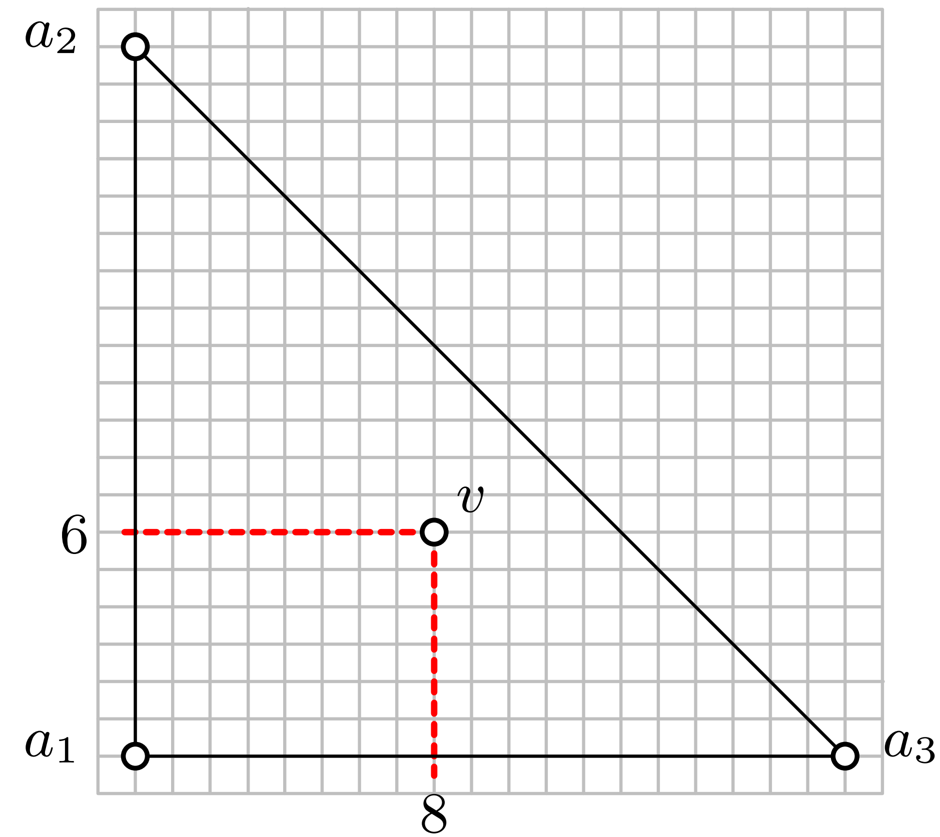


Algo :

- Put the vertices a_1, a_2 et a_3 at positions $(0,0)$, $(0,|F(M)|)$, $(|F(M)|,0)$.
- Put each inner vertex v at position $(|F(R_3(v))|, |F(R_2(v))|)$.

Application to straight-line drawing.



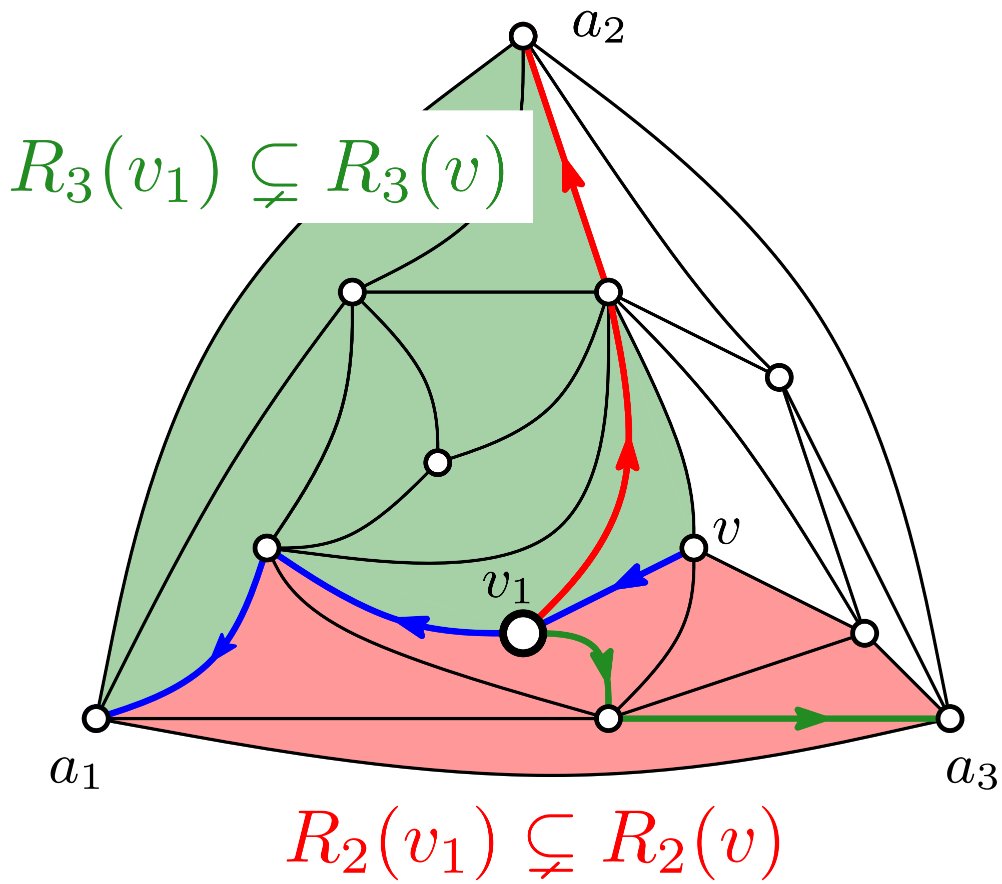


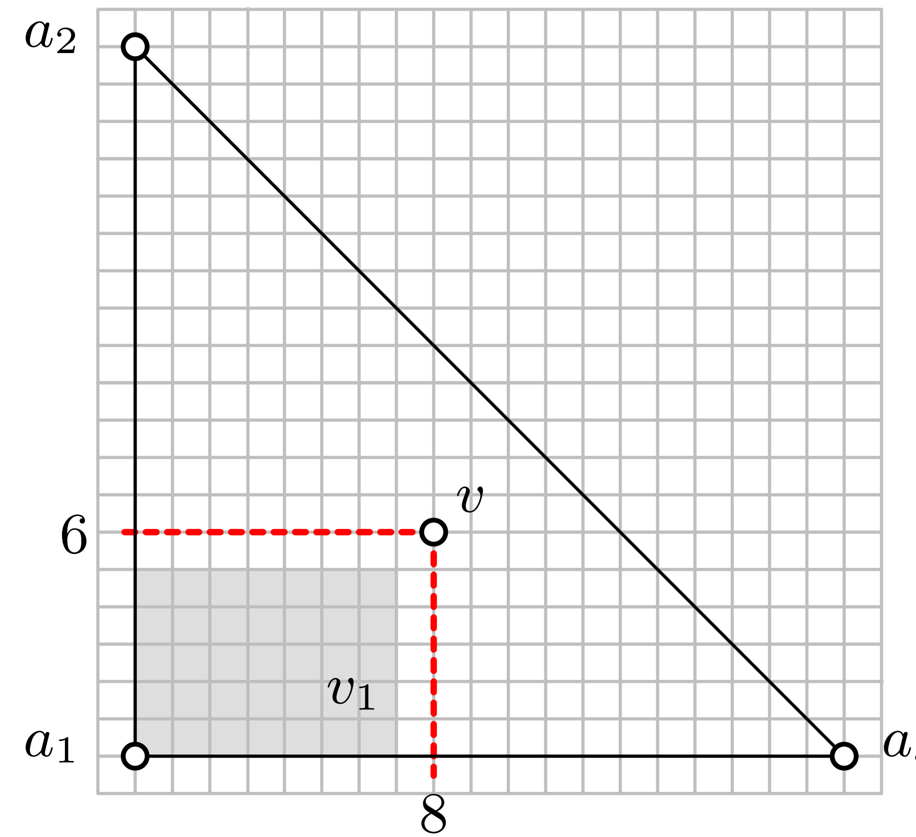
Algo :

- Put the vertices a_1, a_2 et a_3 at positions $(0,0)$, $(0,|F(M)|)$, $(|F(M)|,0)$.
- Put each inner vertex v at position $(|F(R_3(v))|, |F(R_2(v))|)$.

Why is it working ?

Application to straight-line drawing.

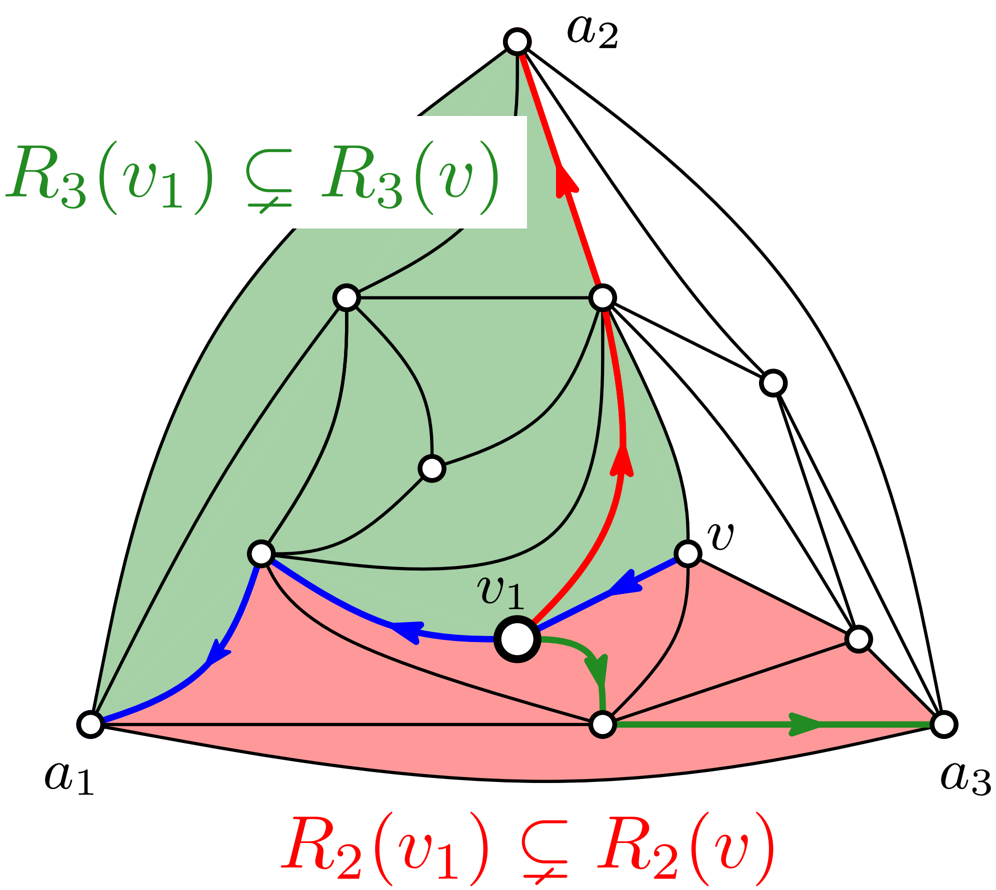


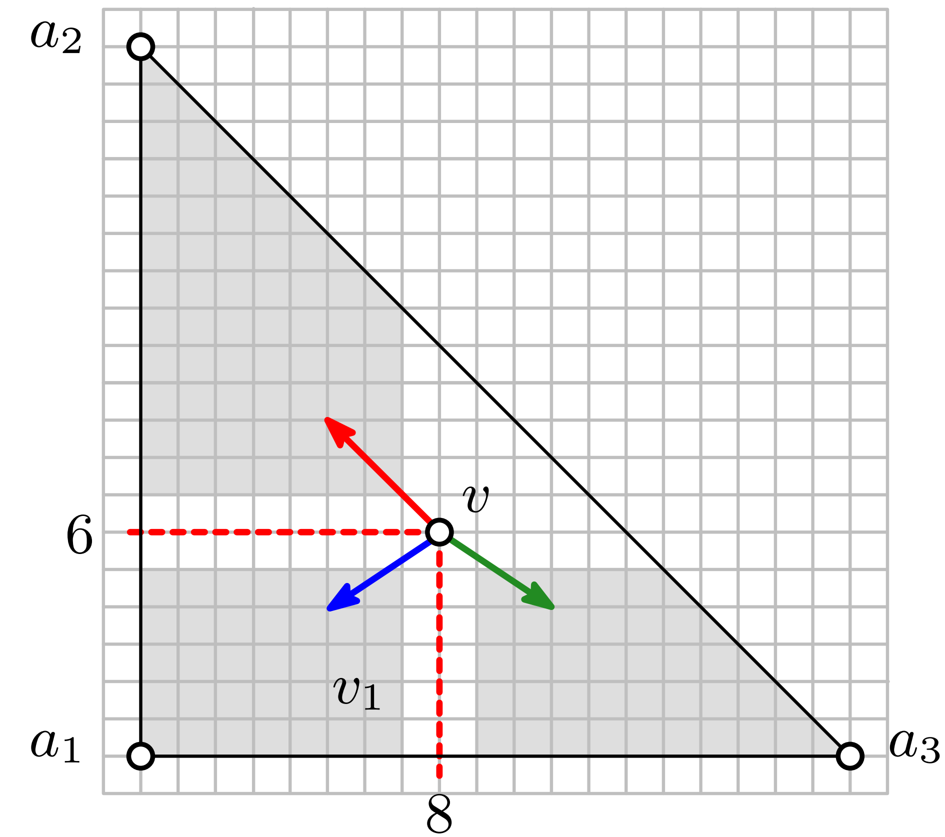


Algo :
- Put the vertices a_1, a_2 et a_3 at positions $(0,0)$, $(0,|F(M)|)$, $(|F(M)|,0)$.
- Put each inner vertex v at position $(|F(R_3(v))|, |F(R_2(v))|)$.

Why is it working ?

Application to straight-line drawing.



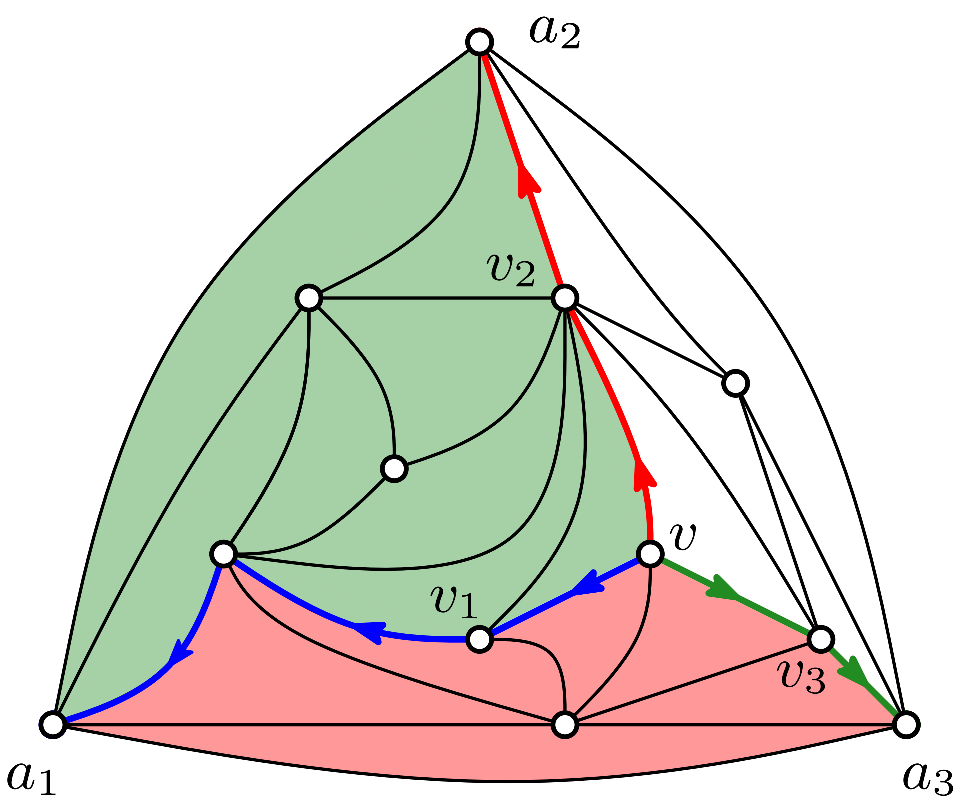


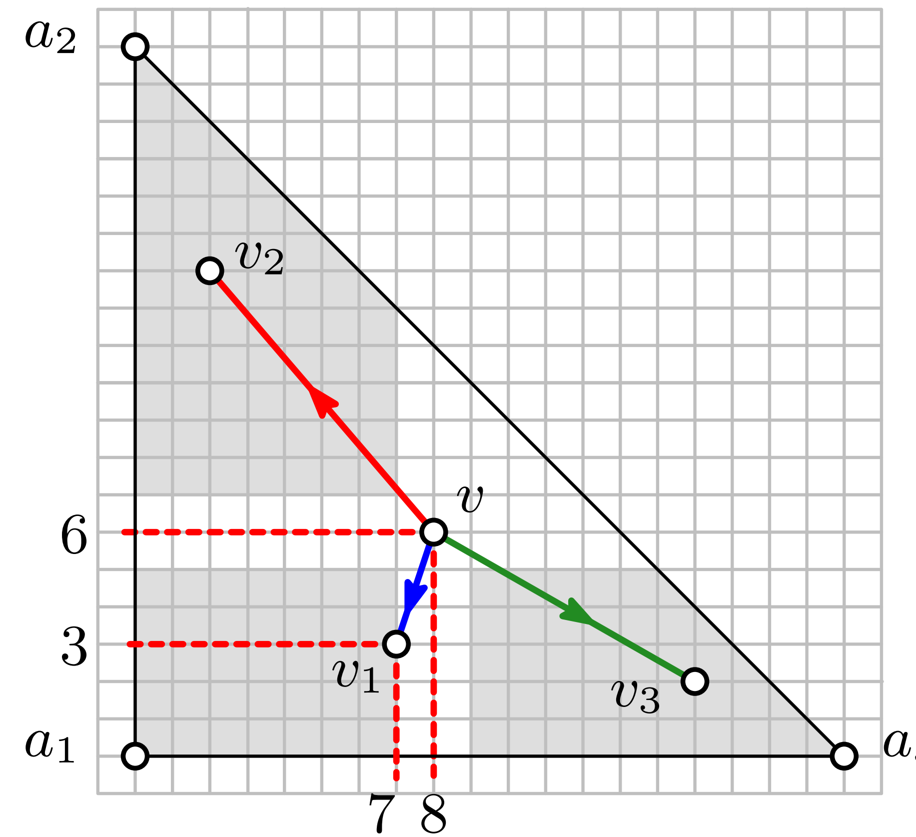
Algo :

- Put the vertices a_1, a_2 et a_3 at positions $(0,0)$, $(0,|F(M)|)$, $(|F(M)|,0)$.
- Put each inner vertex v at position $(|F(R_3(v))|, |F(R_2(v))|)$.

Why is it working ?

Application to straight-line drawing.



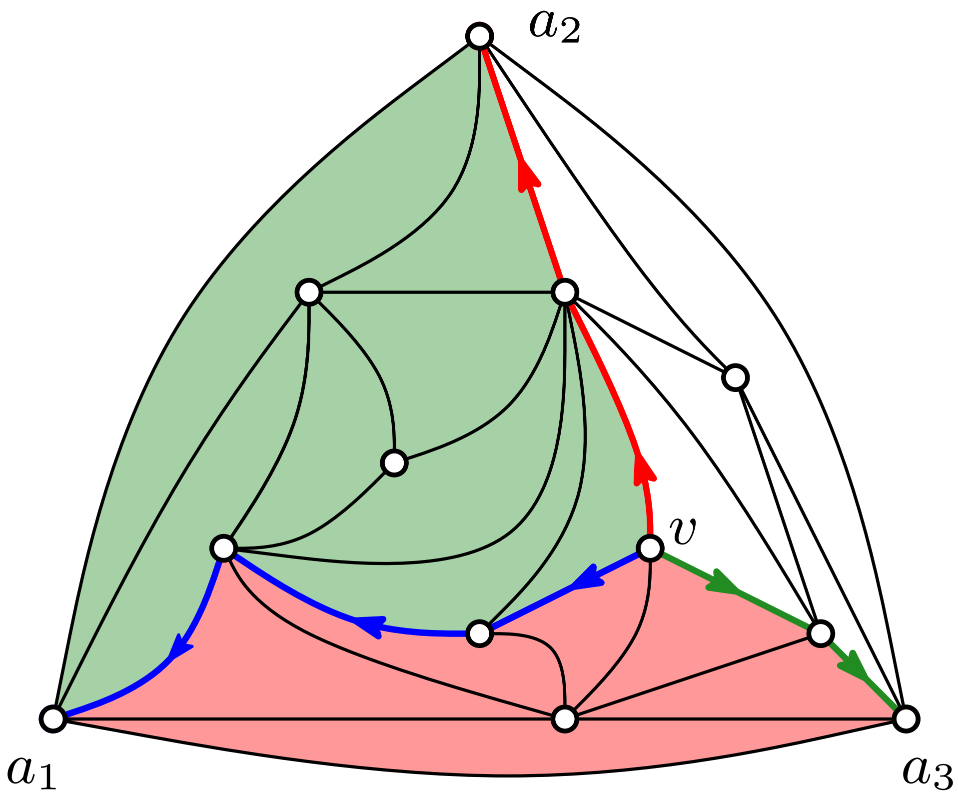


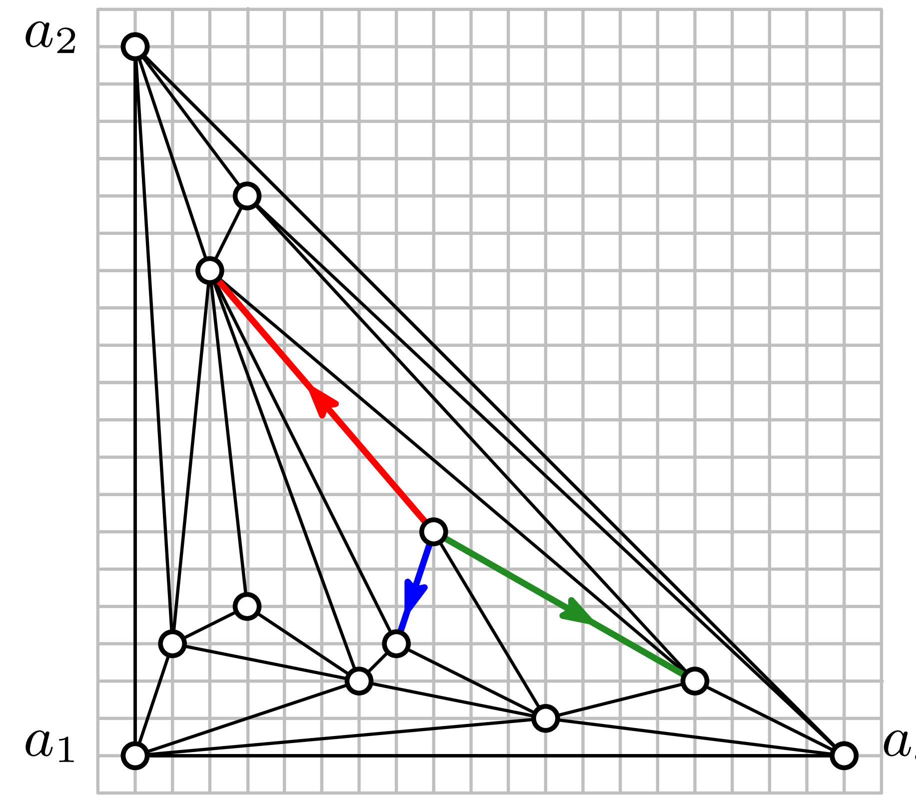
Algo :

- Put the vertices a_1, a_2 et a_3 at positions $(0, 0)$, $(0, |F(M)|)$, $(|F(M)|, 0)$.
- Put each inner vertex v at position $(|F(R_3(v))|, |F(R_2(v))|)$.

Why is it working ?

Application to straight-line drawing.

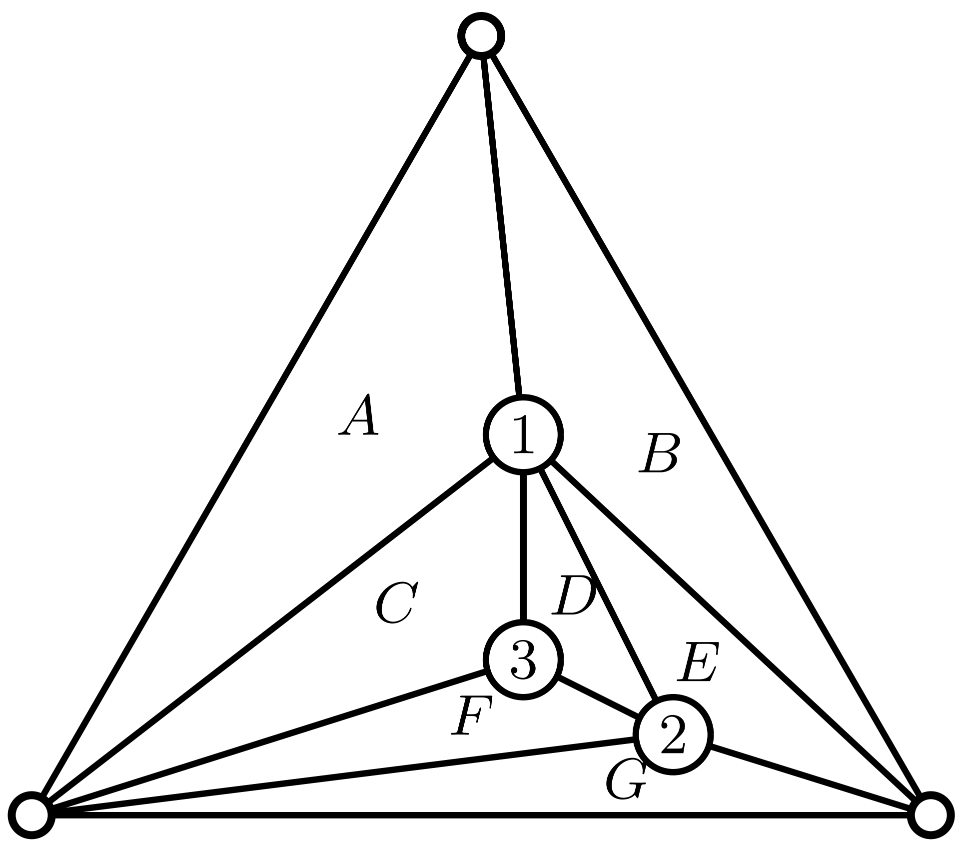


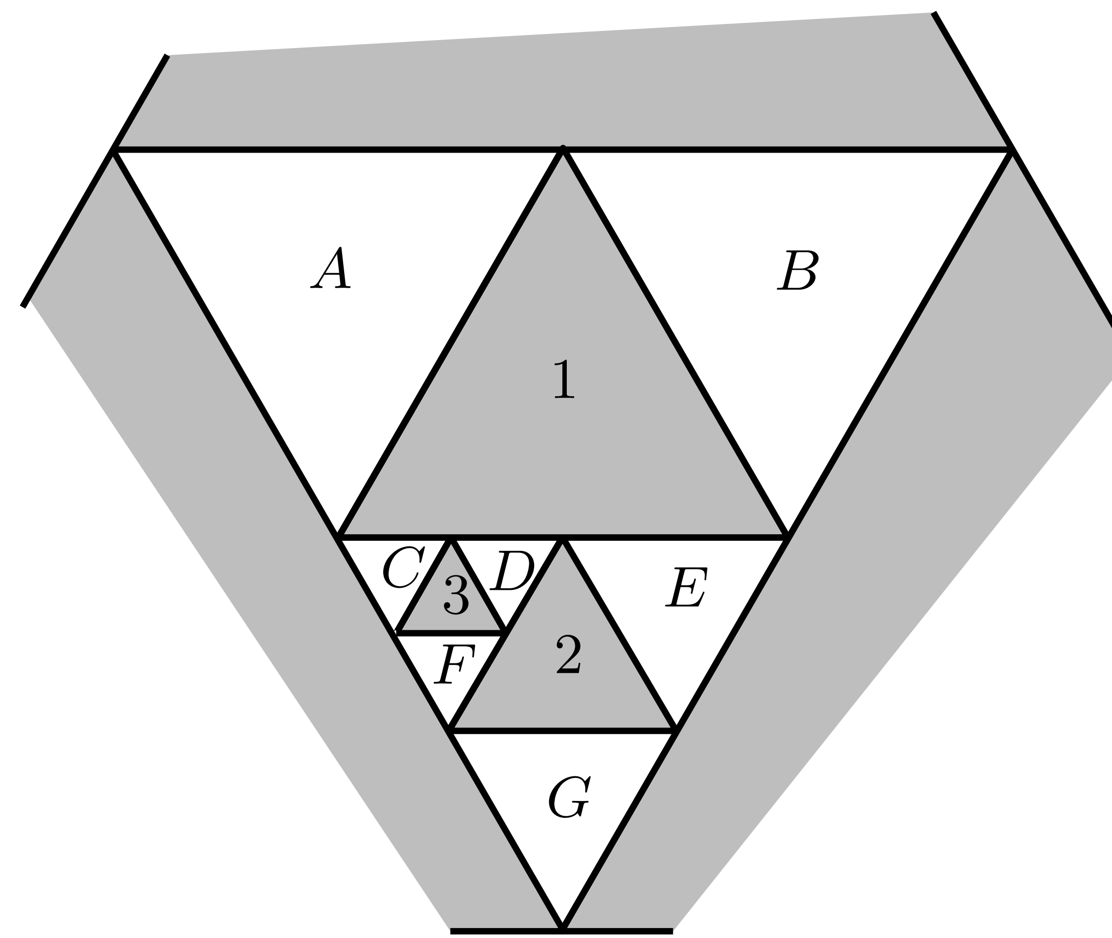


Theorem ([Schnyder '89]) :

This algorithm produces a straightline drawing of the triangulation where all the vertices belong to a grid of size $|F(M)| \times |F(M)|$.

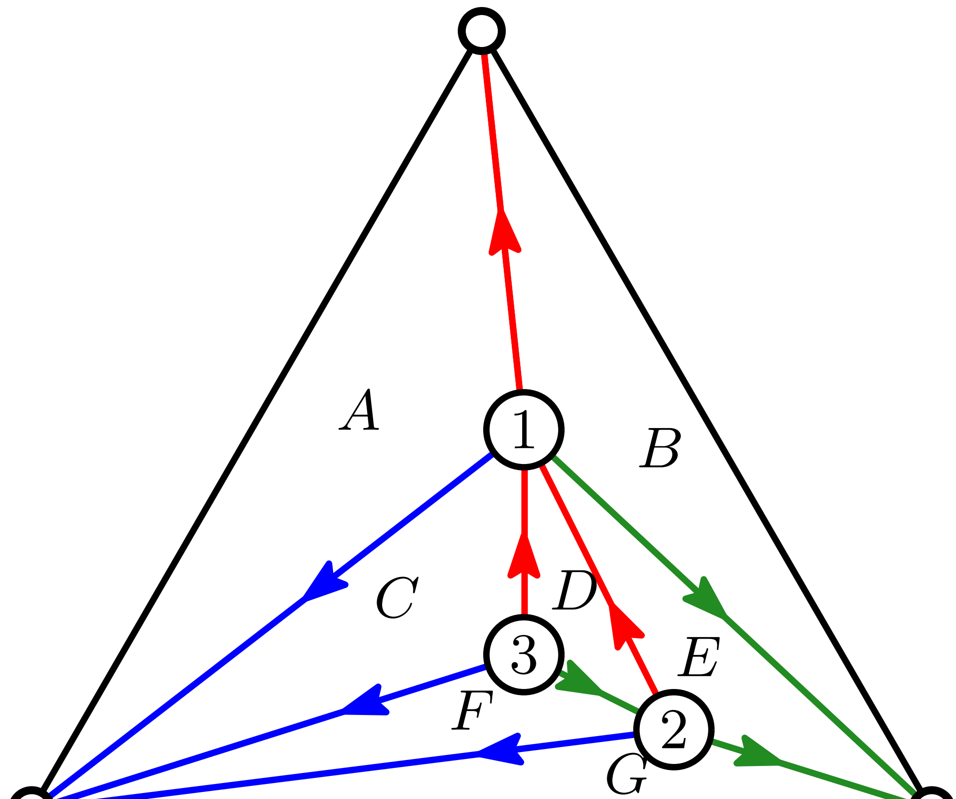
Representation by homothetic triangles.

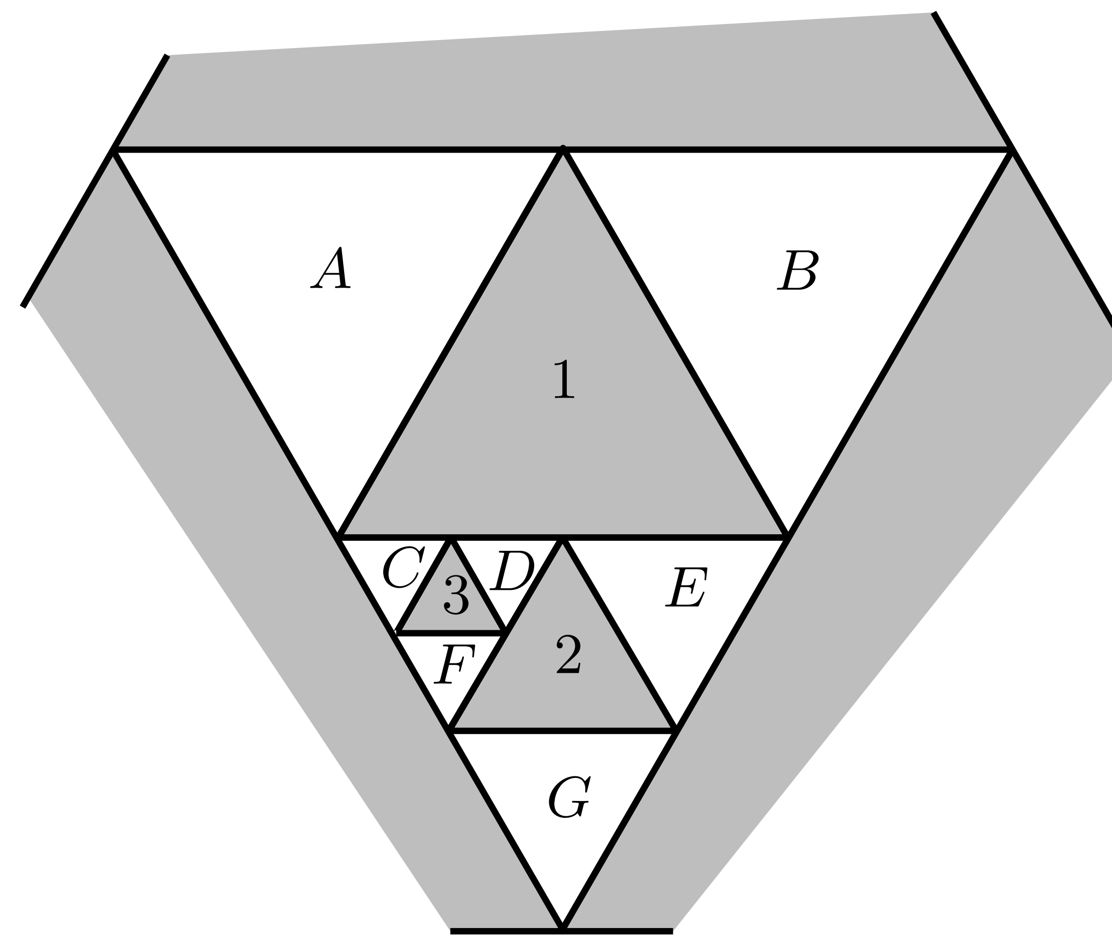




[Motivated by some ideas of Felsner]

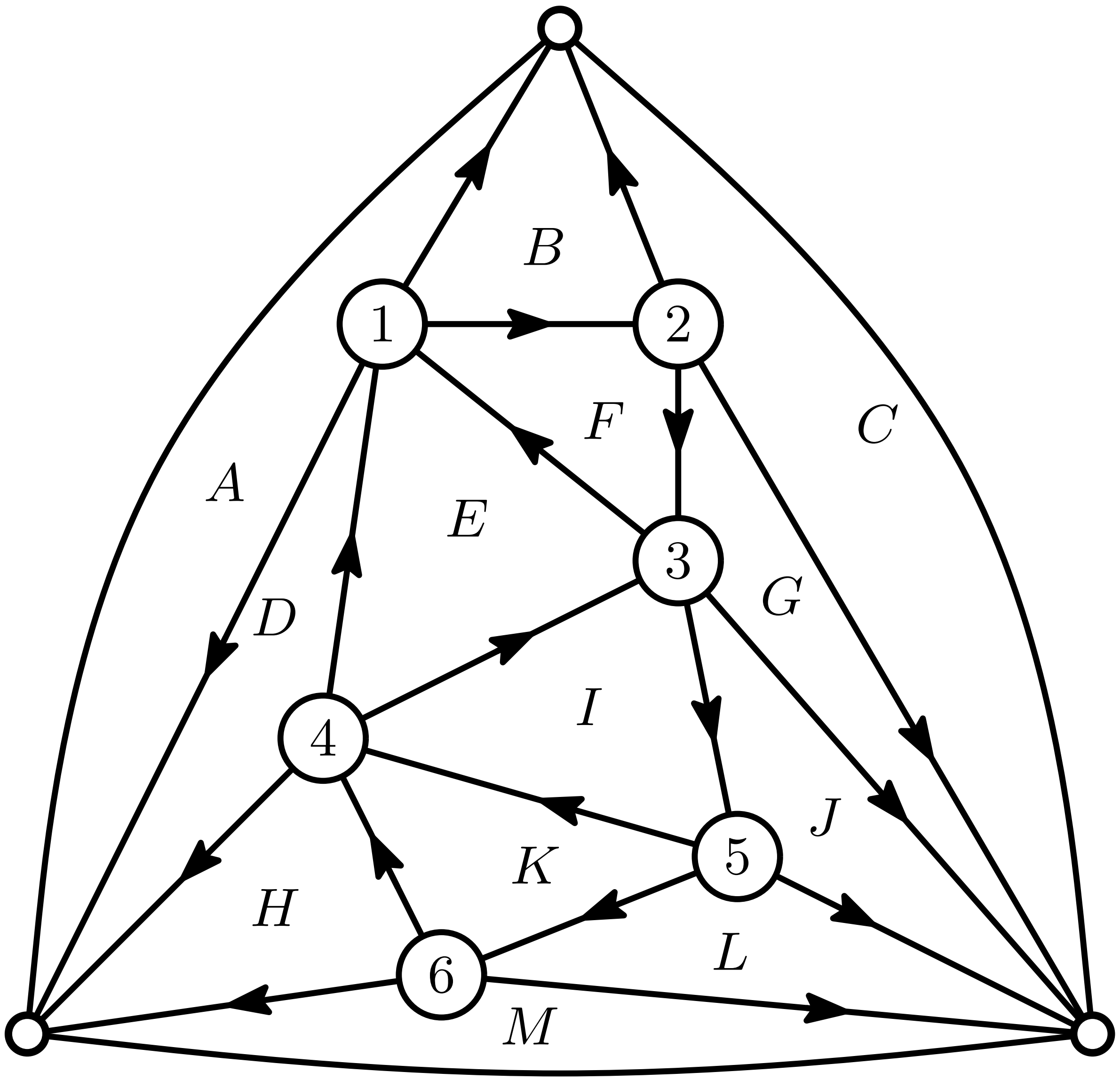
Representation by homothetic triangles.





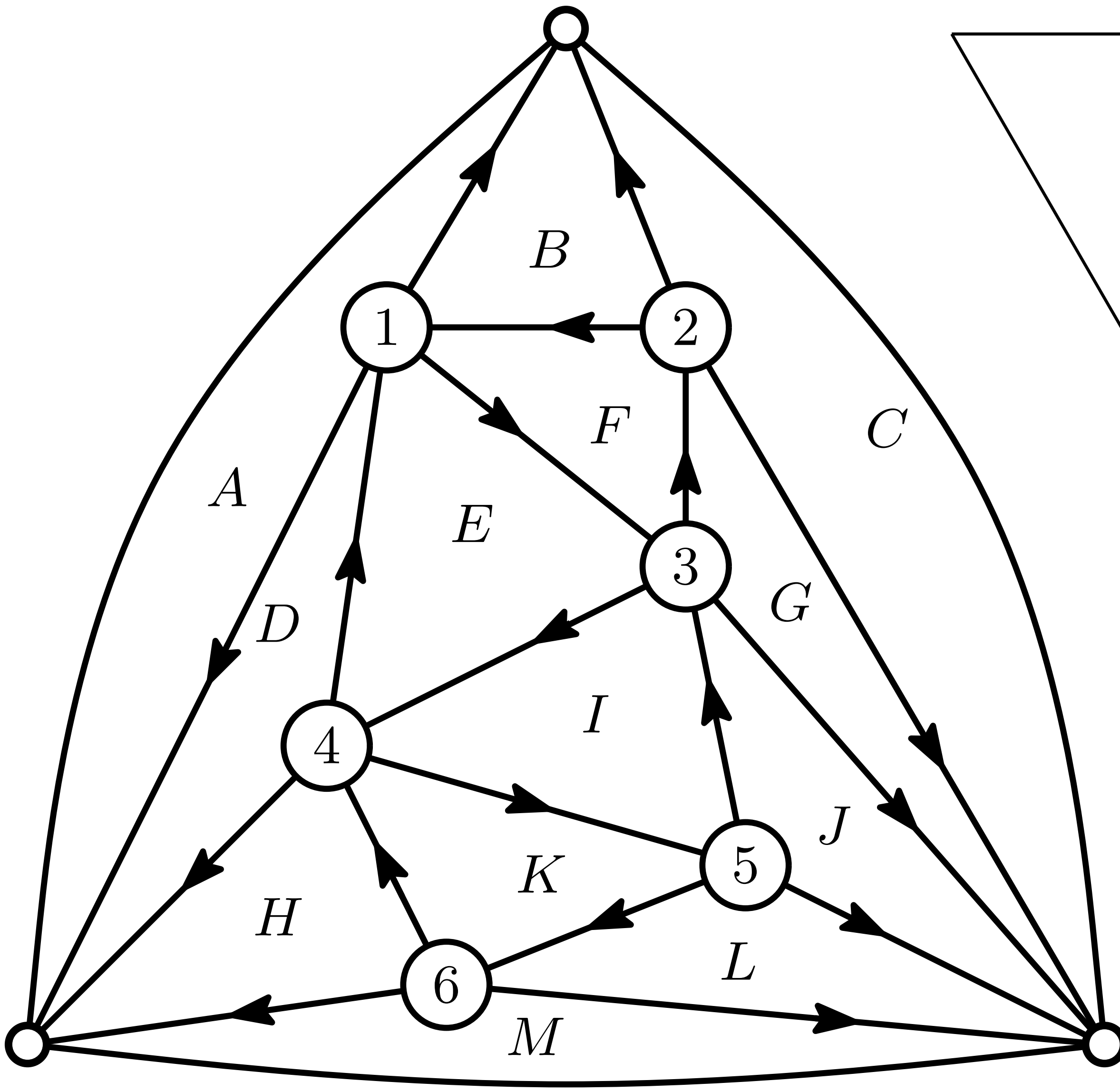
[Motivated by some ideas of Felsner]

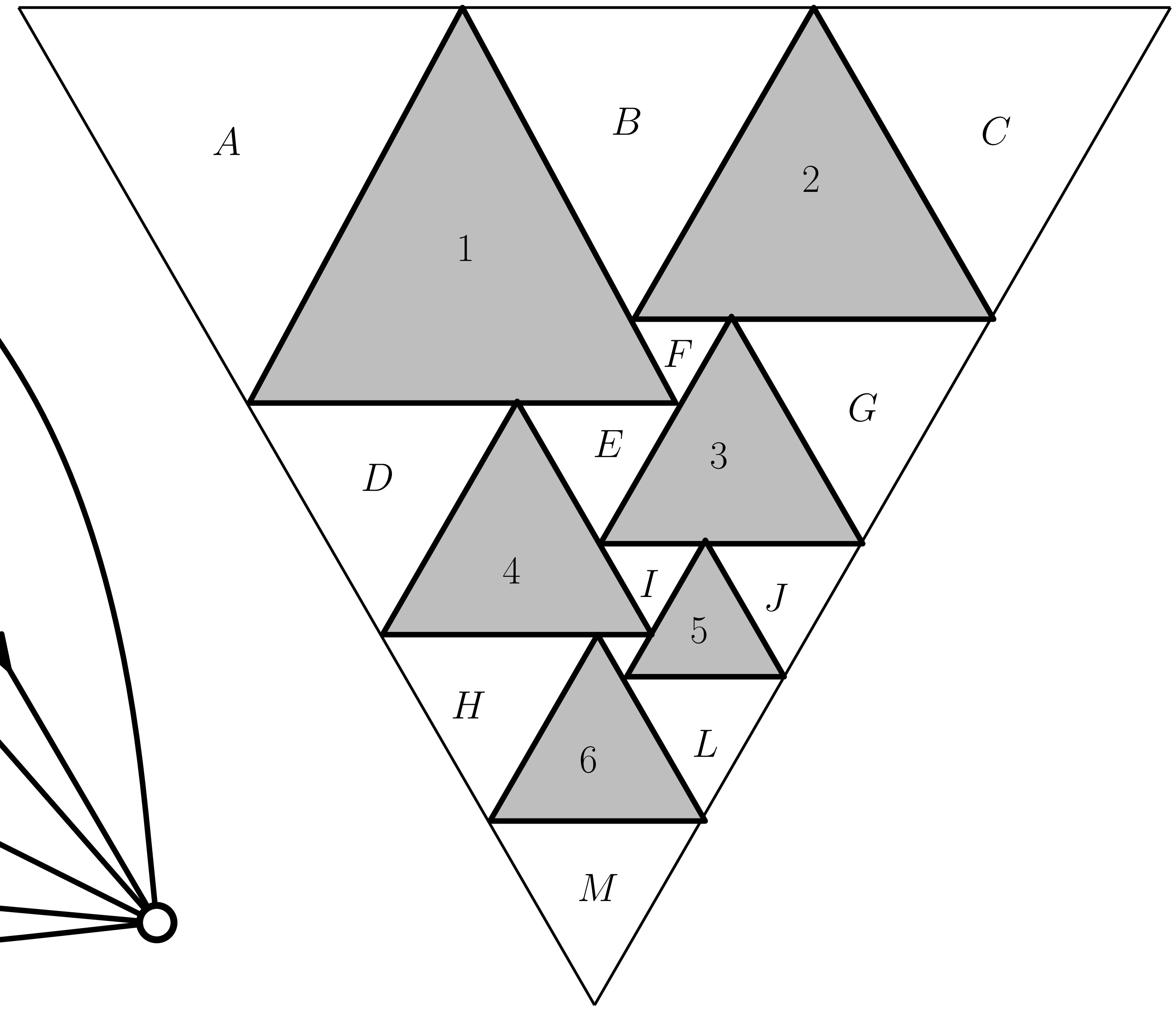
Representation by homothetic triangles.



[Motivated by some ideas of Felsner]

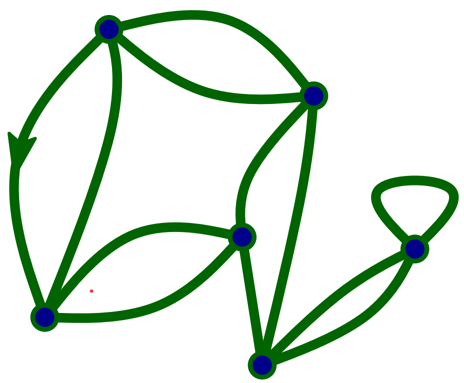
Representation by homothetic triangles.



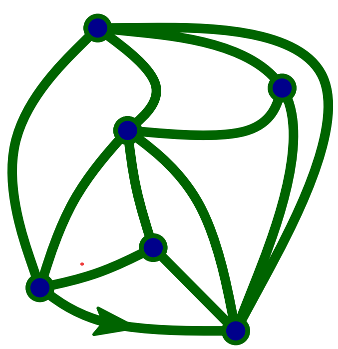


[Motivated by some ideas of Felsner]

Maps enumeration

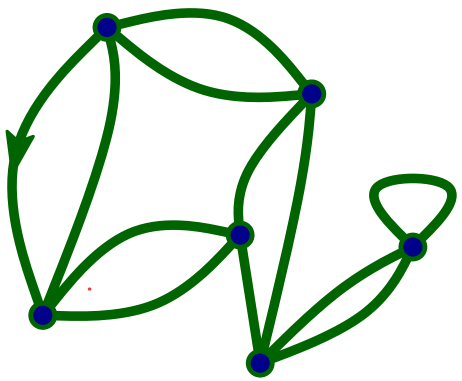


4-regular maps



Simple triangulations (neither loops nor multiple edges)

Maps enumeration

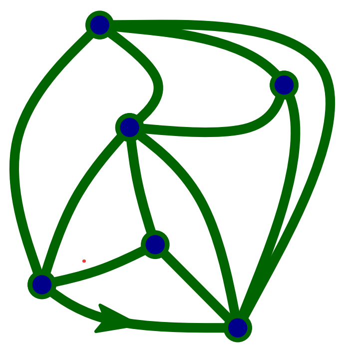


4-regular maps

Number of rooted 4-regular planar maps with n vertices :

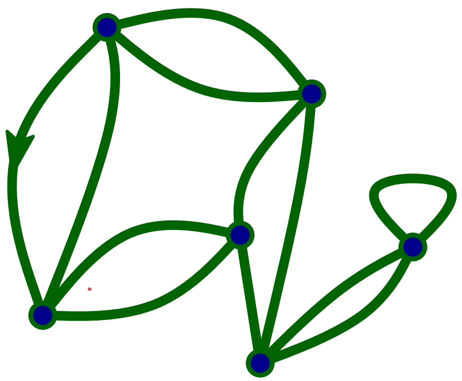
$$R_n = \frac{2}{n+2}\frac{3^n}{n+1}\binom{2n}{n}$$

[Tutte, 62], [Schaeffer '97]



Simple triangulations (neither loops nor multiple edges)

Maps enumeration

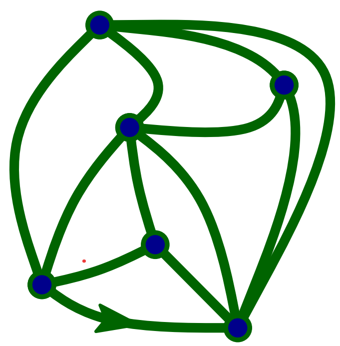


4-regular maps

Number of rooted 4-regular planar maps with n vertices :

$$R_n = \frac{2}{n+2}\frac{3^n}{n+1}\binom{2n}{n}$$

[Tutte, 62], [Schaeffer '97]



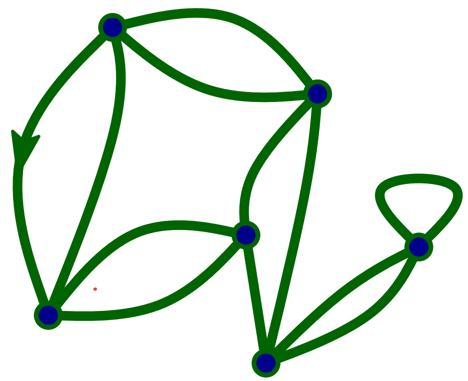
Simple triangulations (neither loops nor multiple edges)

Number of simple triangulations with $n+2$ vertices :

$$\Delta_n = \frac{2\cdot(4n-3)!}{n!(3n-1)!}$$

[Tutte, 62],
[Poulalhon-Schaeffer '05]

Maps enumeration

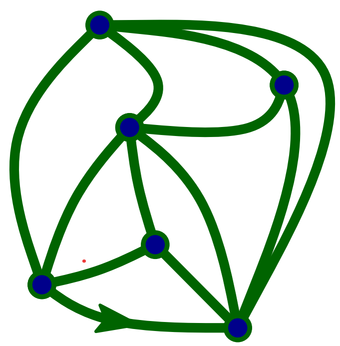


4-regular maps

Number of rooted 4-regular planar maps with n vertices :

$$R_n = \frac{2}{n+2}\frac{3^n}{n+1}\binom{2n}{n}$$

[Tutte, 62], [Schaeffer '97]



Simple triangulations (neither loops nor multiple edges)

Number of simple triangulations with $n+2$ vertices :

$$\Delta_n = \frac{2 \cdot (4n-3)!}{n!(3n-1)!}$$

[Tutte, 62],
[Poulalhon-Schaeffer '05]

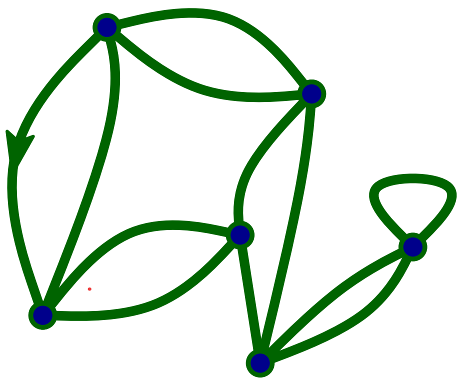
Many methods to count maps :

Recursive decomposition [Tutte '60s + ...], matrix integrals [t'Hooft '74 + ...]

Bijective proofs [Cori-Vauquelin-Schaeffer, Bouttier-diFrancesco-Guitter, Bernardi, Fusy, Poulalhon, ...]

= bijections btw maps and labeled trees or btw maps and blossoming trees.

Maps enumeration

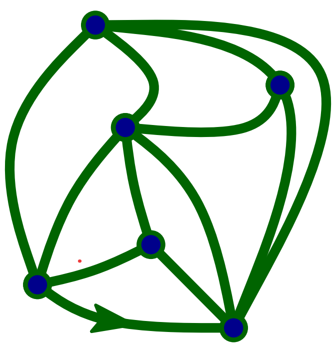


4-regular maps

Number of rooted 4-regular planar maps with n vertices :

$$R_n = \frac{2}{n+2} \frac{3^n}{n+1} \binom{2n}{n}$$

[Tutte, 62], [Schaeffer '97]



Simple triangulations (neither loops nor multiple edges)

Number of simple triangulations with $n+2$ vertices :

$$\Delta_n = \frac{2 \cdot (4n-3)!}{n!(3n-1)!}$$

[Tutte, 62],
[Poulalhon-Schaeffer '05]

Many methods to count maps :

Recursive decomposition [Tutte '60s + ...], matrix integrals [t'Hooft '74 + ...]

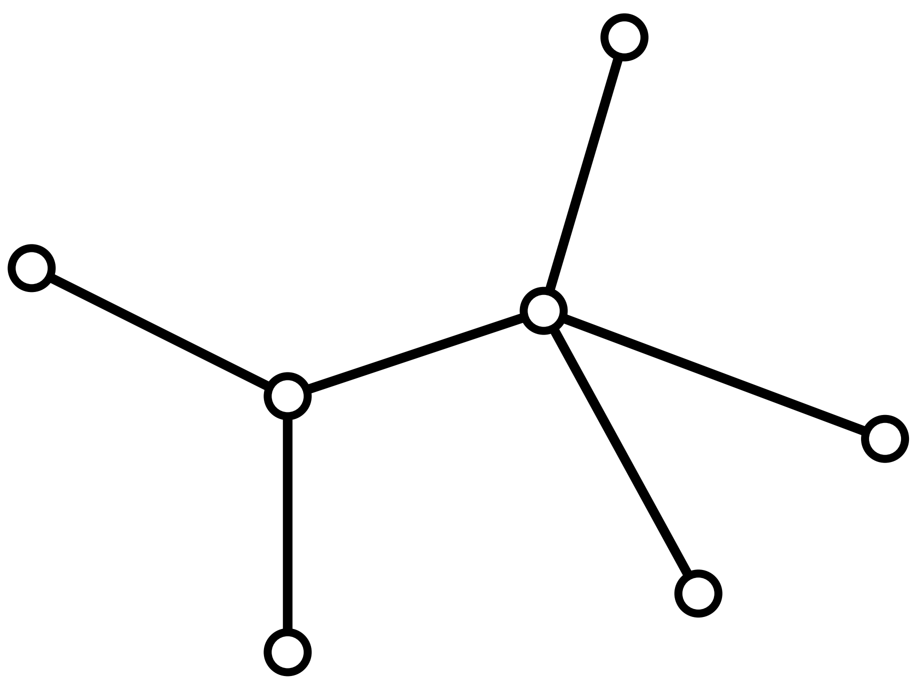
Bijective proofs [Cori-Vauquelin-Schaeffer, Bouttier-diFrancesco-Guitter, Bernardi, Fusy, Poulalhon, ...]

= bijections btw maps and labeled trees or btw maps and blossoming trees.

What is a blossoming tree ?

A **blossoming tree** is a plane tree whose vertices can carry **opening stems** or **closing stems** (or both) such that :

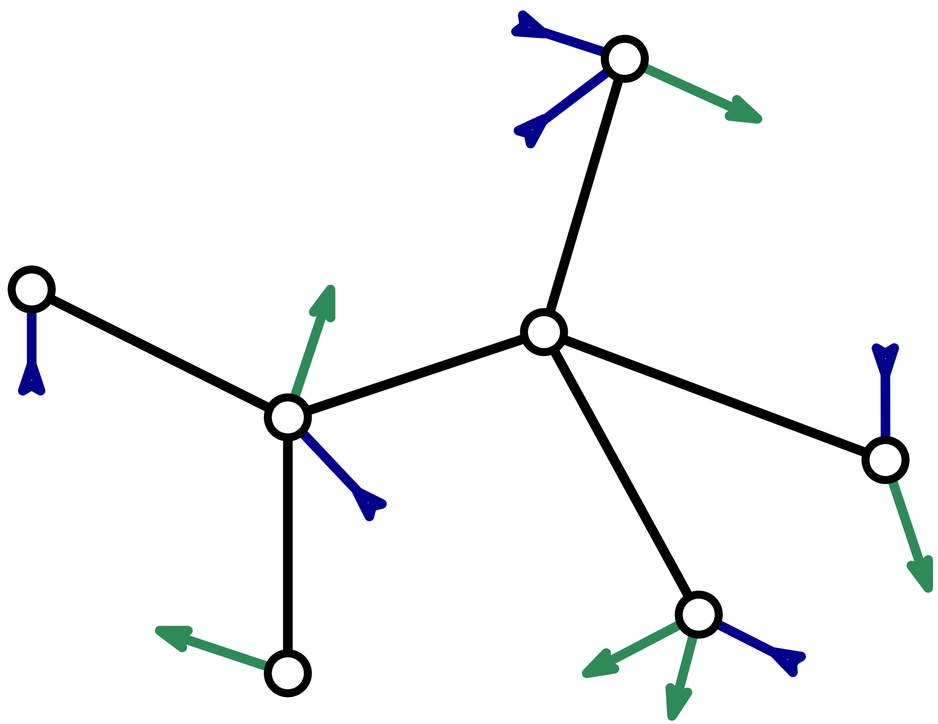
$$\#\ \text{opening stems} = \#\ \text{closing stems}$$



What is a blossoming tree ?

A **blossoming tree** is a plane tree whose vertices can carry **opening stems** or **closing stems** (or both) such that :

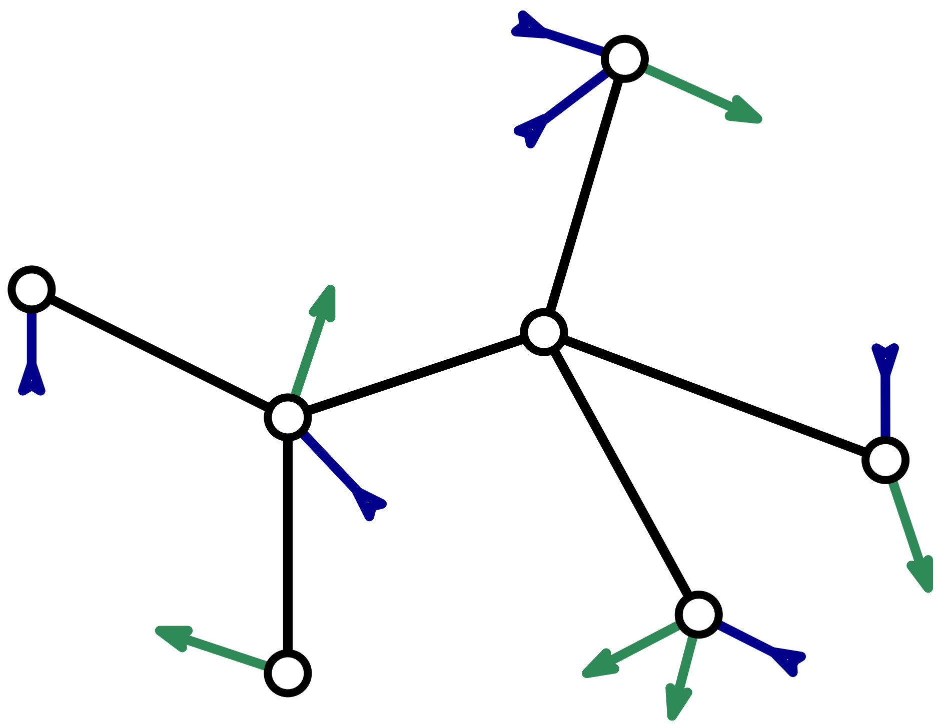
$$\# \text{ opening stems} = \# \text{ closing stems}$$

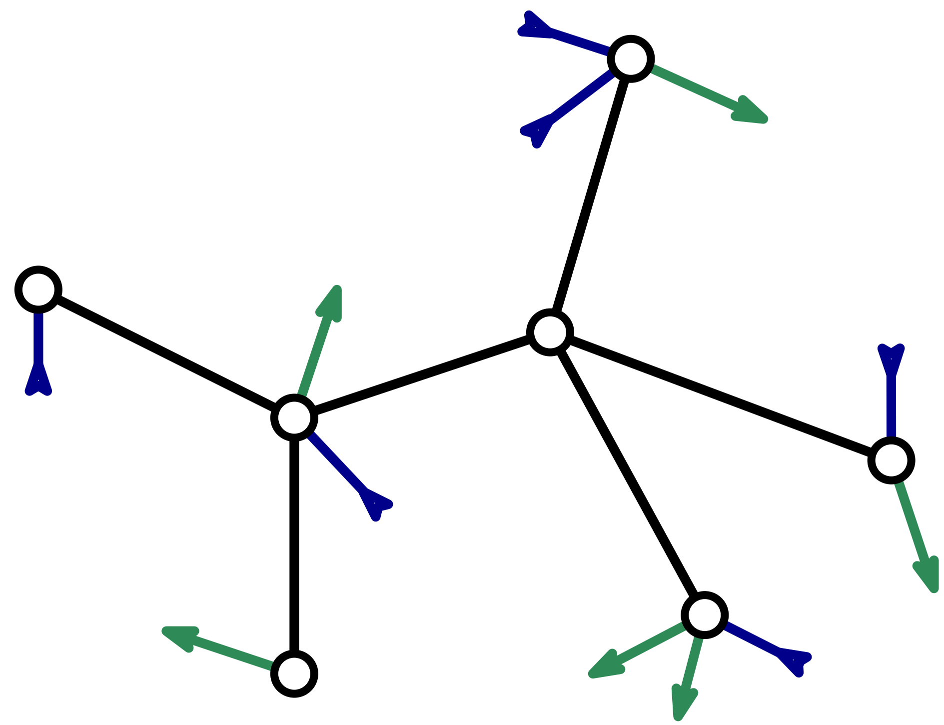


What is a blossoming tree ?

A **blossoming tree** is a plane tree whose vertices can carry **opening stems** or **closing stems** (or both) such that :

$$\# \text{ opening stems} = \# \text{ closing stems}$$

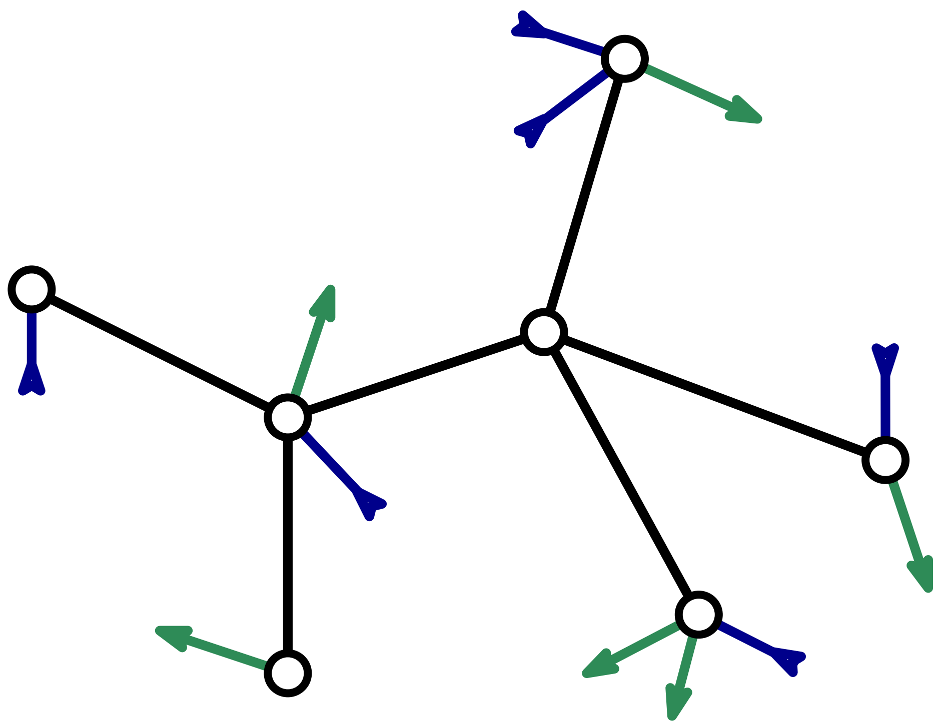


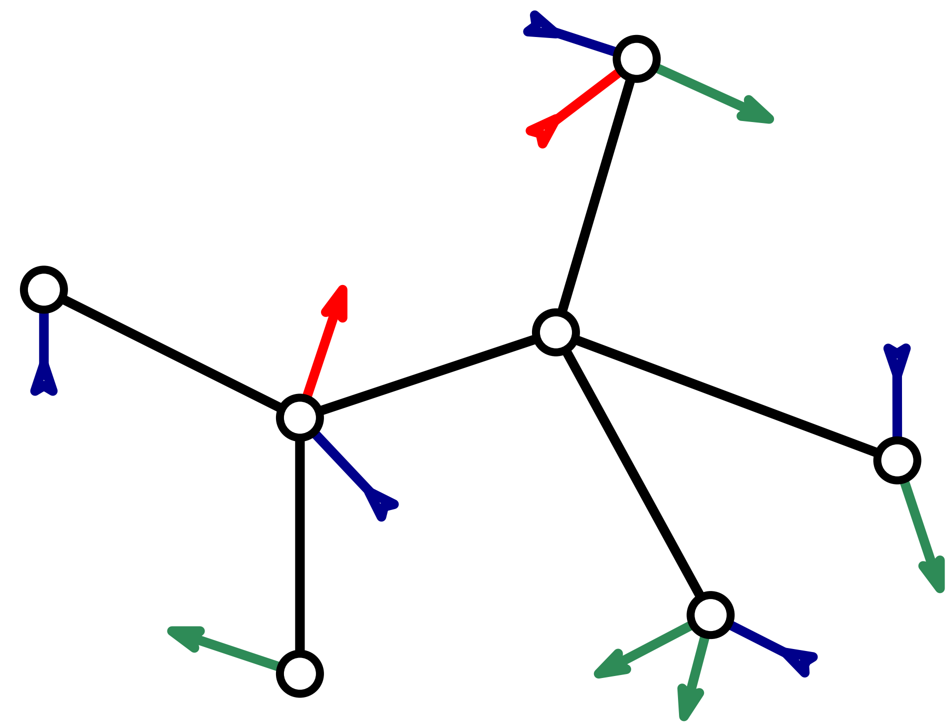


What is a blossoming tree ?

A **blossoming tree** is a plane tree whose vertices can carry **opening stems** or **closing stems** (or both) such that :

$$\# \text{ opening stems} = \# \text{ closing stems}$$

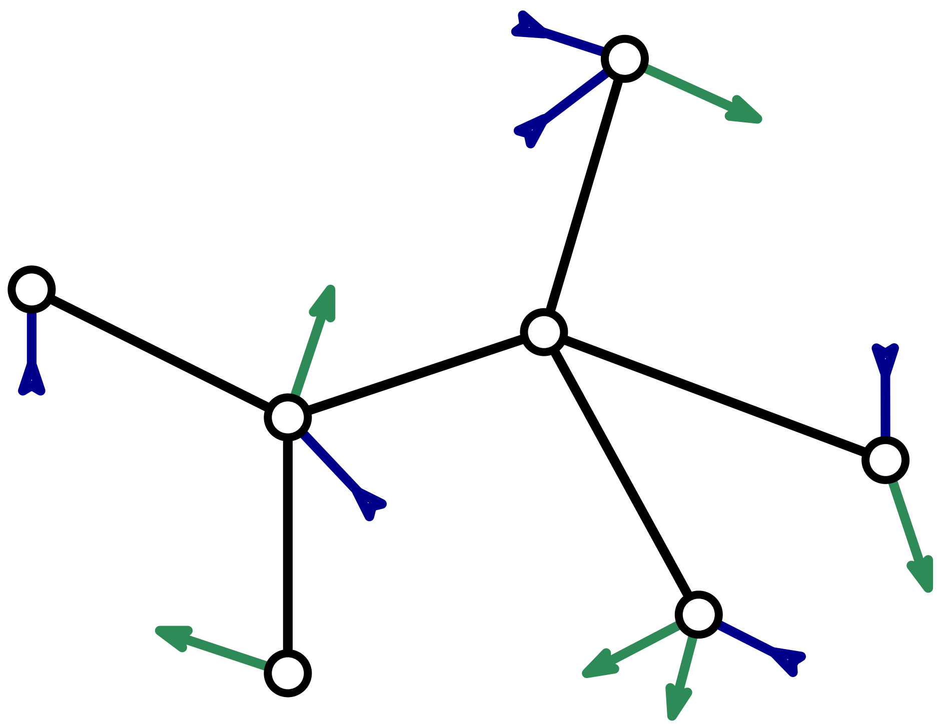


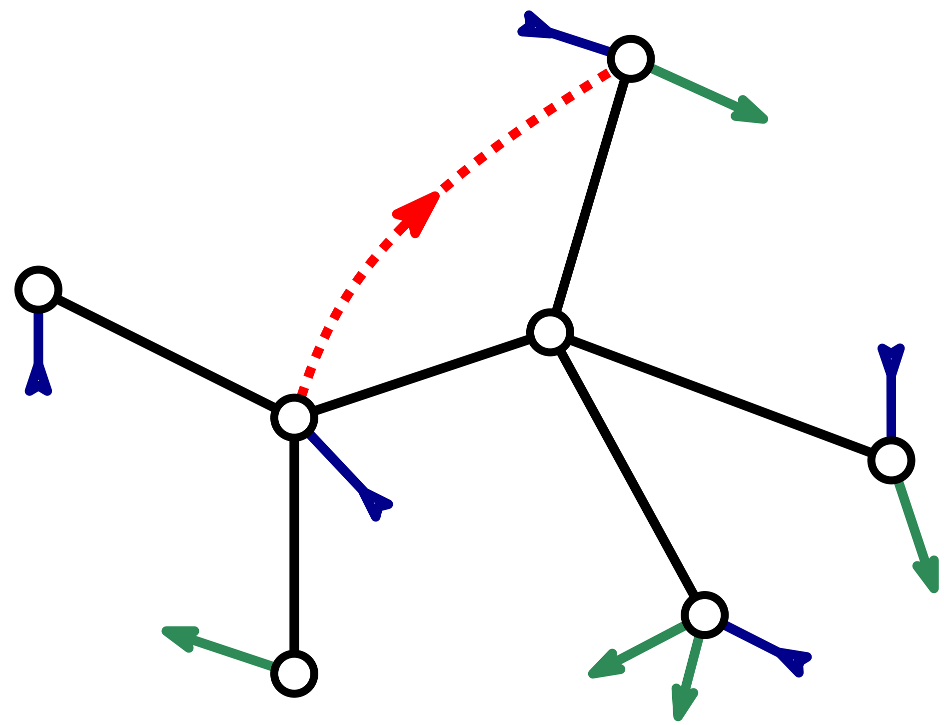


What is a blossoming tree ?

A **blossoming tree** is a plane tree whose vertices can carry **opening stems** or **closing stems** (or both) such that :

$$\#\ \text{opening stems} = \#\ \text{closing stems}$$

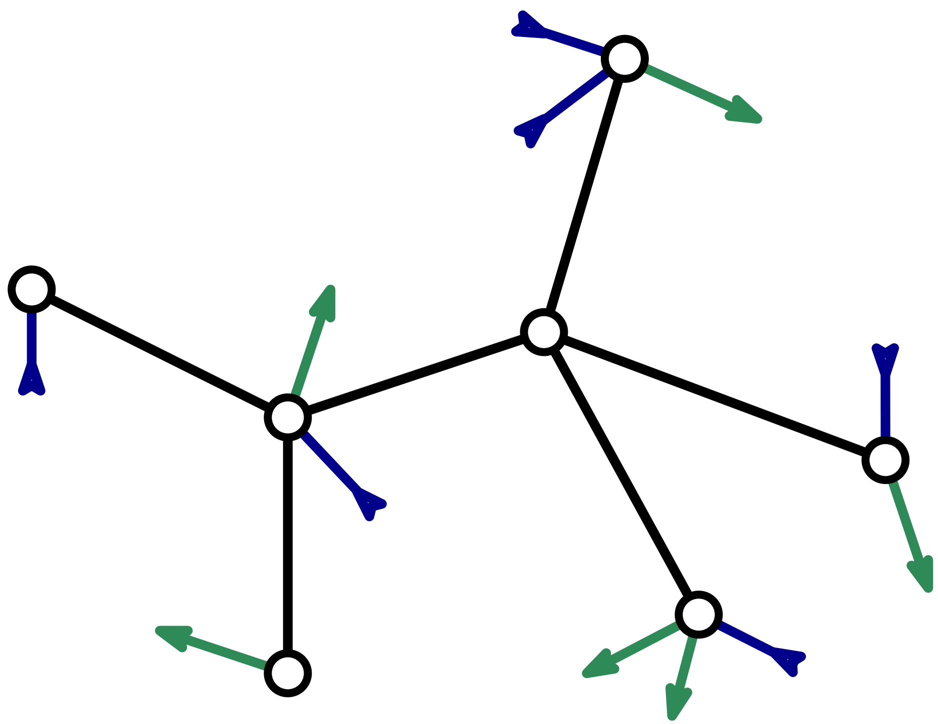


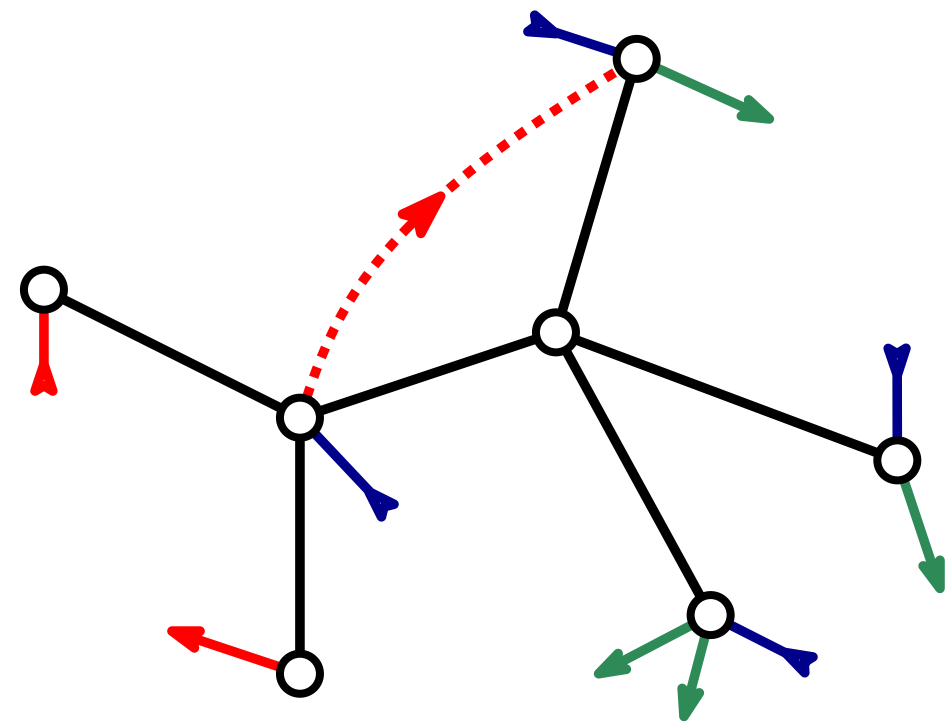


What is a blossoming tree ?

A **blossoming tree** is a plane tree whose vertices can carry **opening stems** or **closing stems** (or both) such that :

$$\# \text{ opening stems} = \# \text{ closing stems}$$

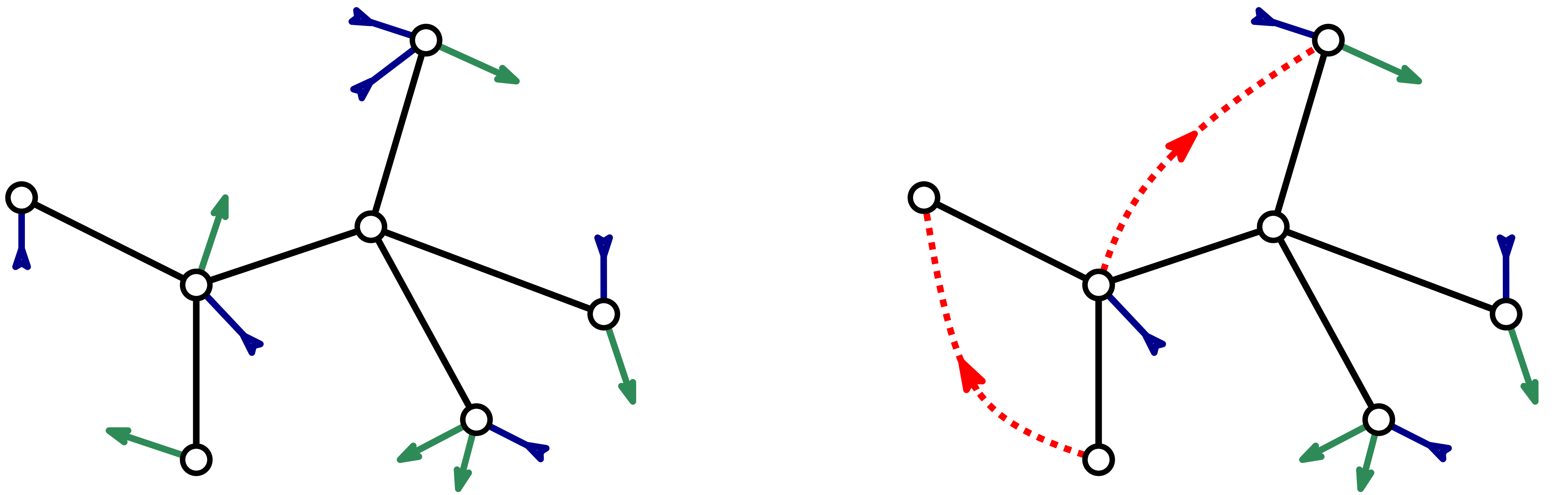




What is a blossoming tree ?

A **blossoming tree** is a plane tree whose vertices can carry **opening stems** or **closing stems** (or both) such that :

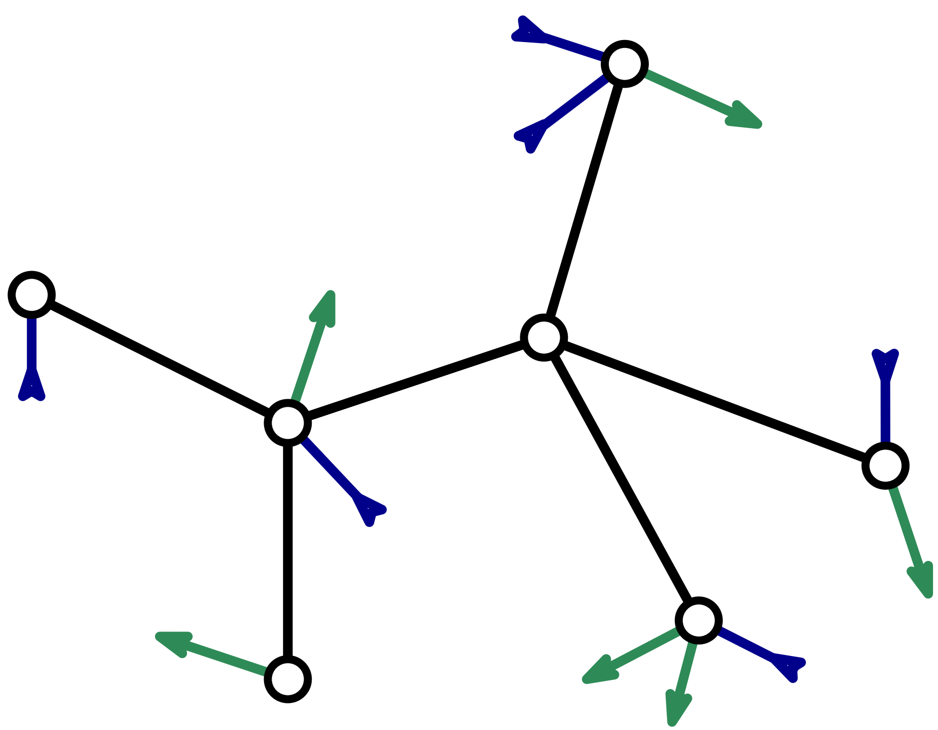
$$\#\text{ opening stems} = \#\text{ closing stems}$$

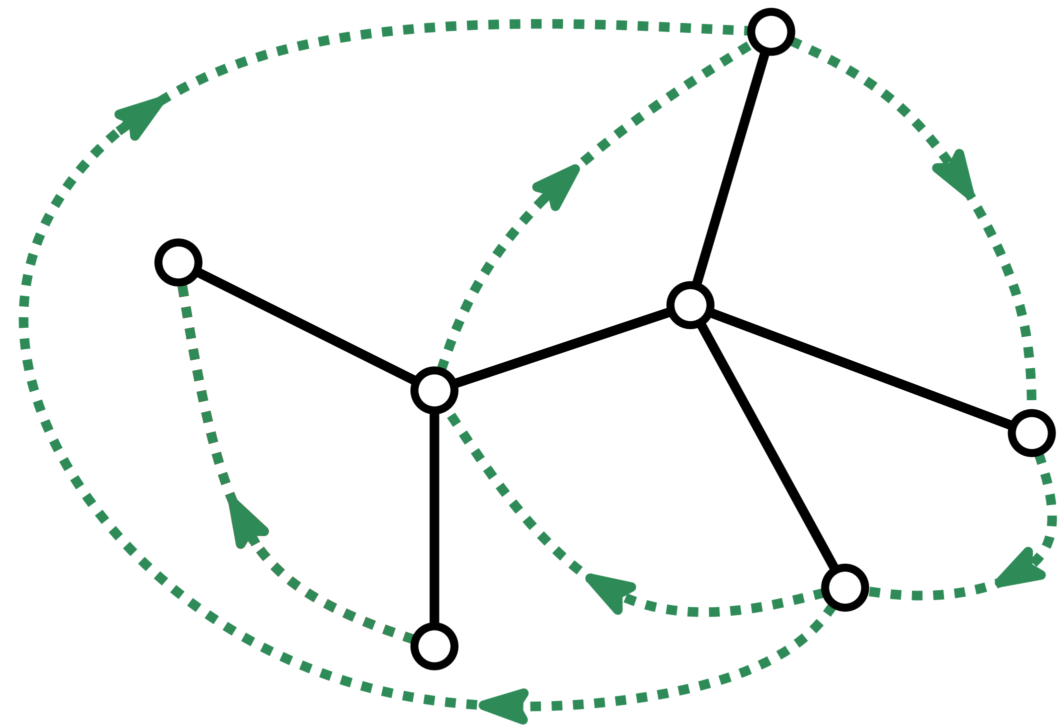


What is a blossoming tree ?

A **blossoming tree** is a plane tree whose vertices can carry **opening stems** or **closing stems** (or both) such that :

$$\# \text{ opening stems} = \# \text{ closing stems}$$

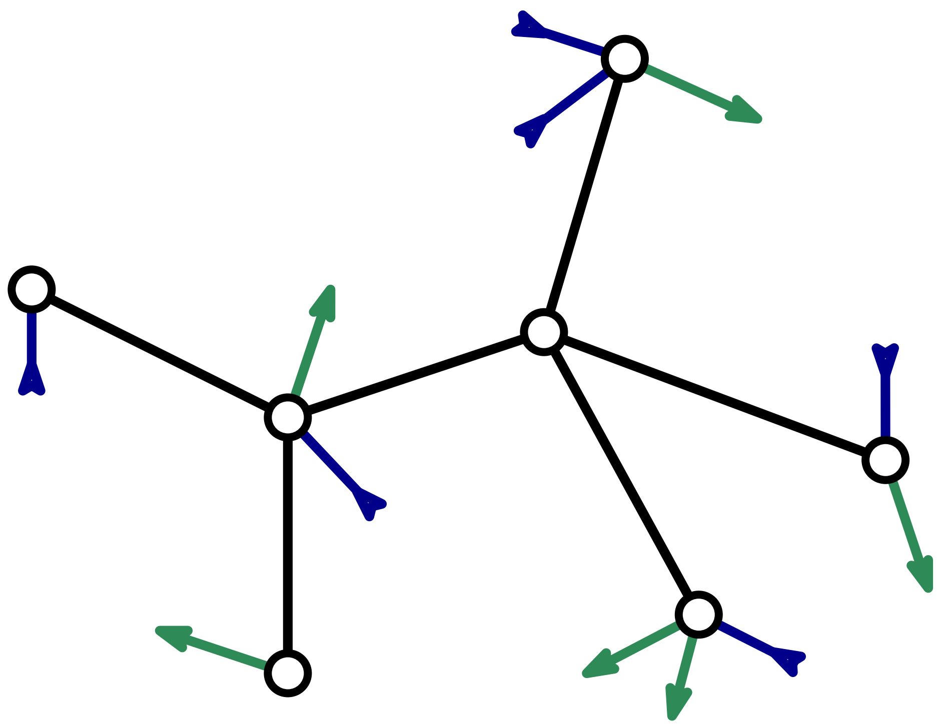


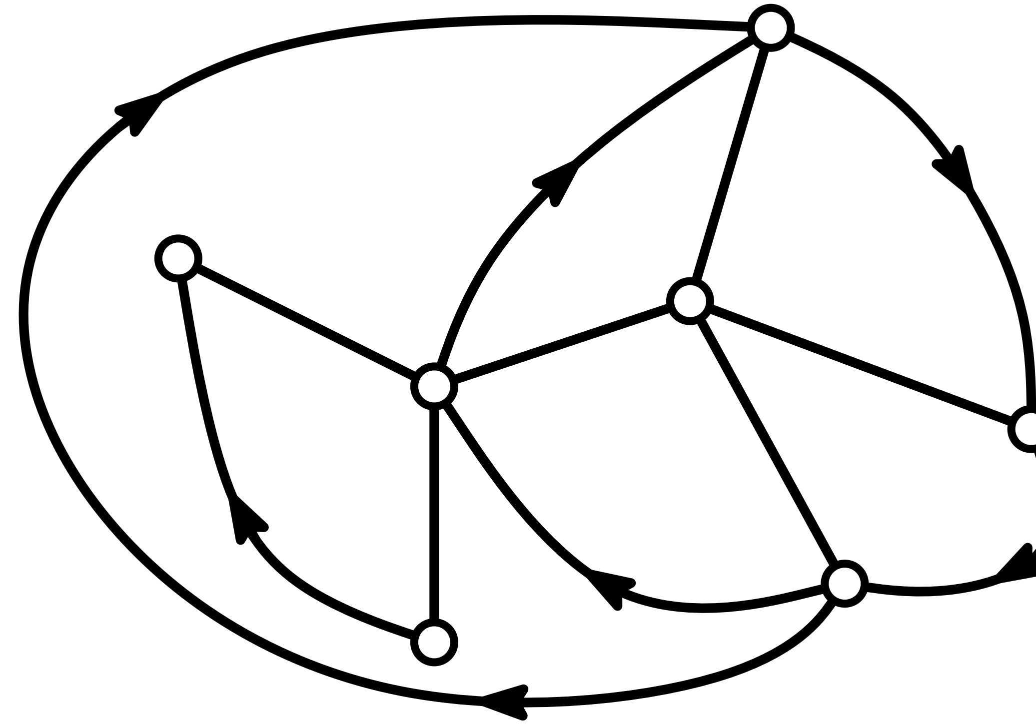


What is a blossoming tree ?

A **blossoming tree** is a plane tree whose vertices can carry **opening stems** or **closing stems** (or both) such that :

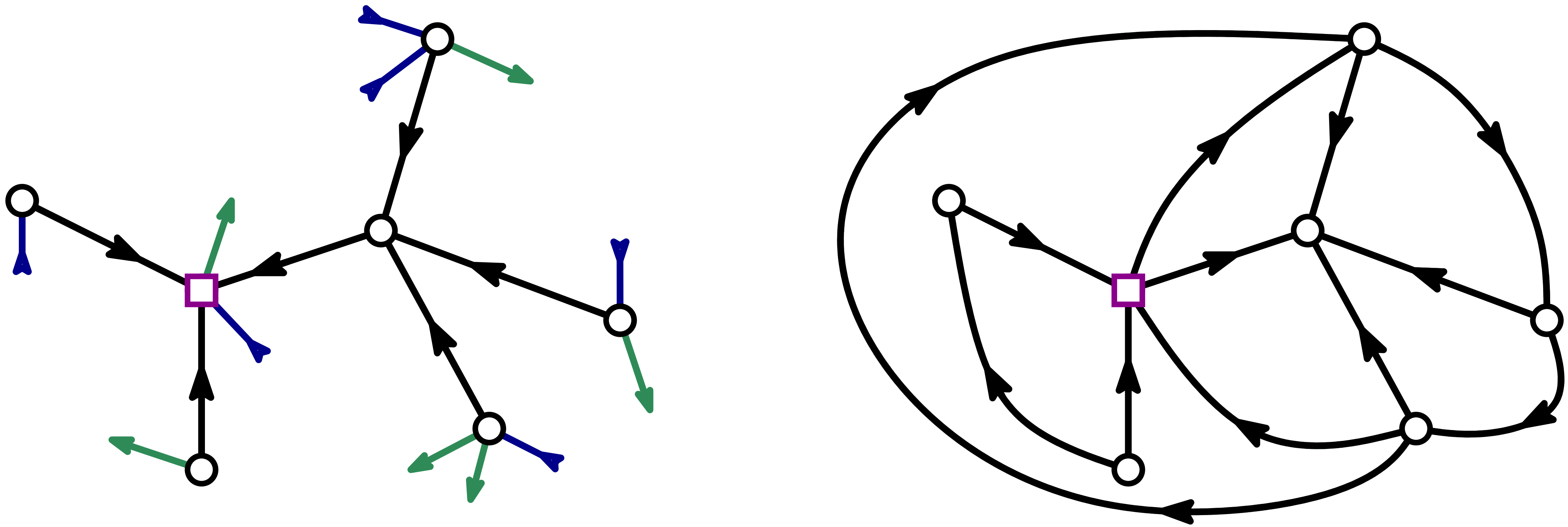
$$\#\ \text{opening stems} = \#\ \text{closing stems}$$





A plane map can canonically associated to any blossoming tree by making all closures clockwise.

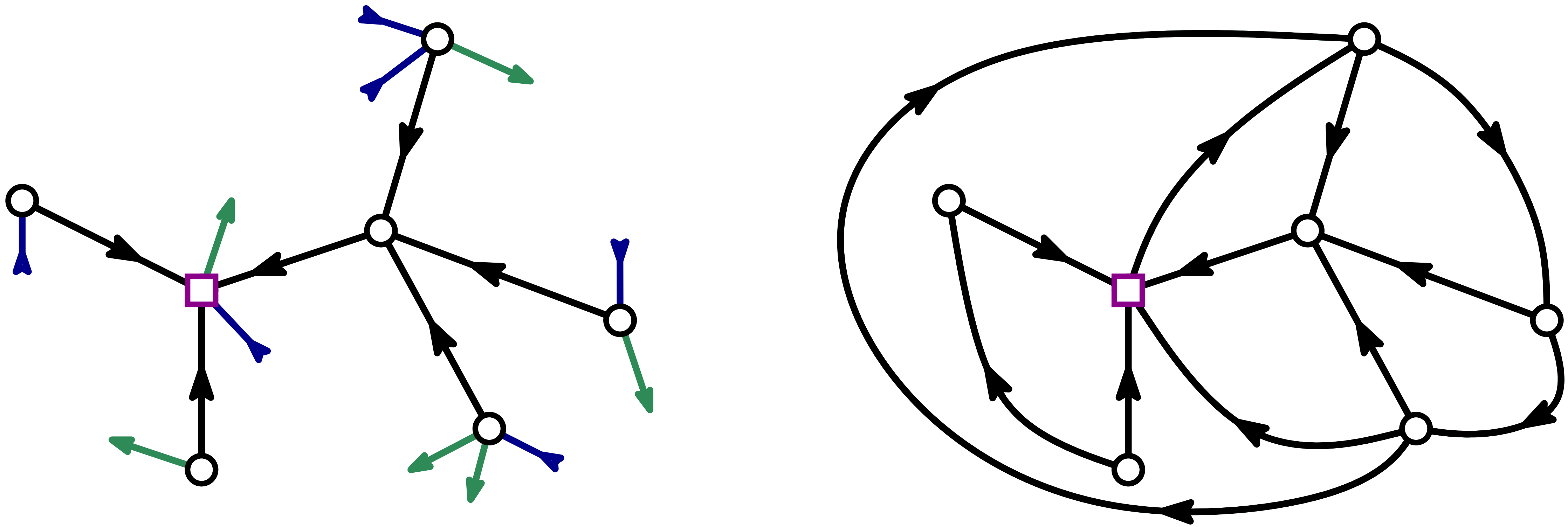
What is a blossoming tree ?



A plane map can be canonically associated to any blossoming tree by making all closures clockwise.

If the edges of the tree are oriented + closure edges oriented naturally
$\Rightarrow$ orientation of the map without ccw cycles.

What is a blossoming tree ?



A plane map can be canonically associated to any blossoming tree by making all closures clockwise.

If the edges of the tree are oriented **towards its root**
$+$ closure edges oriented naturally
$\Rightarrow$ **accessible** orientation of the map without ccw cycles.

Can we always map a map to a blossoming tree ?

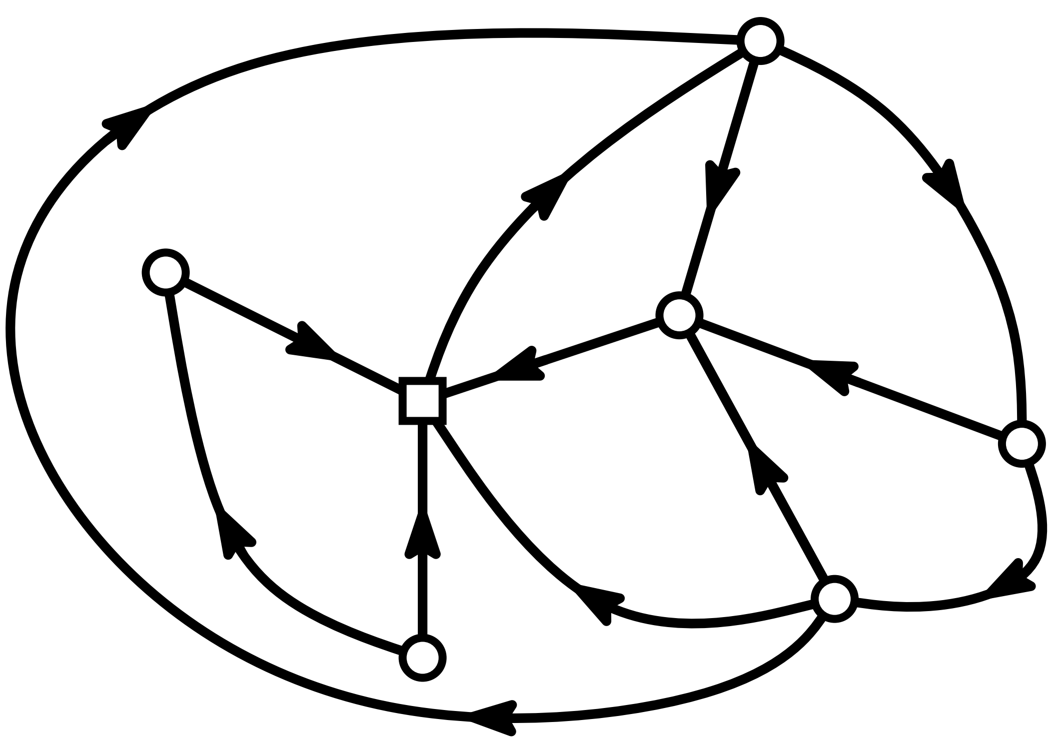
Theorem [Bernardi '07], [A., Poulalhon '14+] :

Let M be a rooted plane map endowed with an **accessible** and **minimal** ($=$ without ccw cycles) orientation.
Then, there exists a **unique** rooted blossoming tree whose closure is M endowed with the same orientation.

Can we always map a map to a blossoming tree ?

Theorem [Bernardi '07], [A., Poulalhon '14+] :

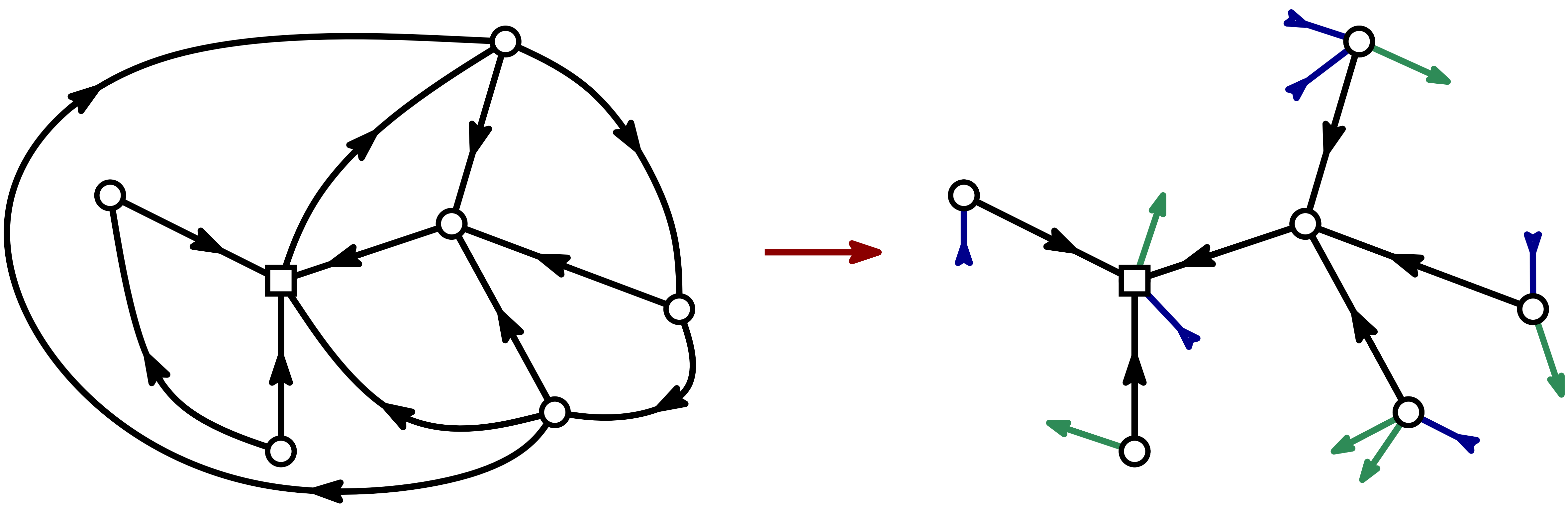
Let M be a rooted plane map endowed with an **accessible** and **minimal** ($=$ without ccw cycles) orientation.
Then, there exists a **unique** rooted blossoming tree whose closure is M endowed with the same orientation.



Can we always map a map to a blossoming tree ?

Theorem [Bernardi '07], [A., Poulalhon '14+] :

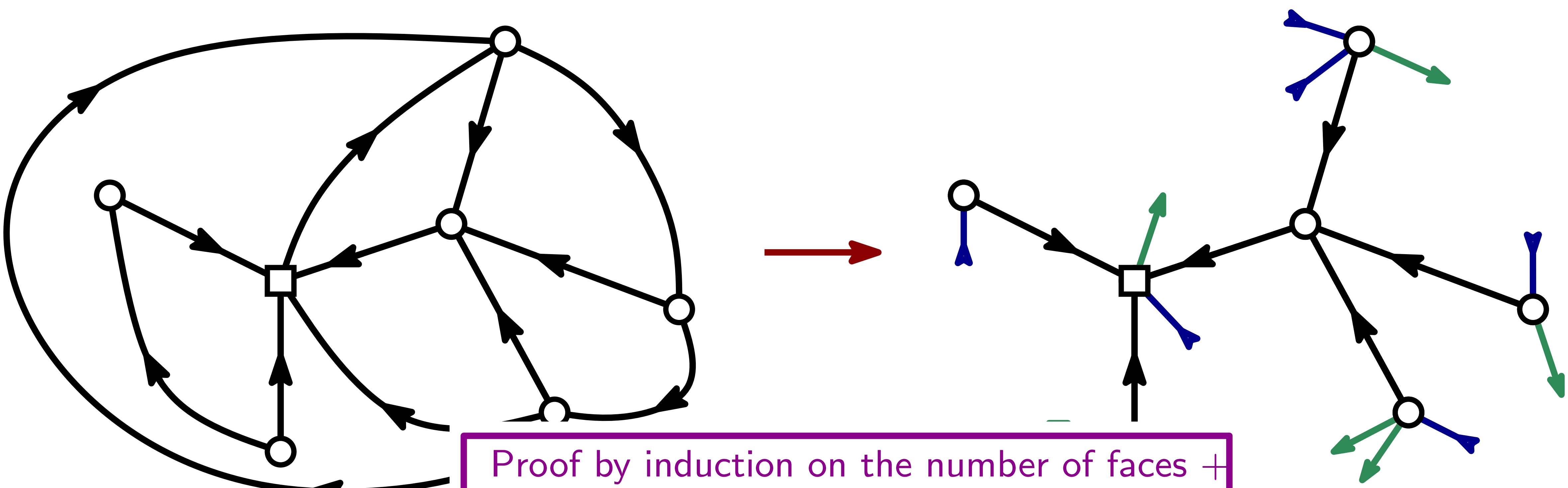
Let M be a rooted plane map endowed with an **accessible** and **minimal** (= without ccw cycles) orientation.
Then, there exists a **unique** rooted blossoming tree whose closure is M endowed with the same orientation.



Can we always map a map to a blossoming tree ?

Theorem [Bernardi '07], [A., Poulalhon '14+] :

Let M be a rooted plane map endowed with an **accessible** and **minimal** ($=$ without ccw cycles) orientation.
Then, there exists a **unique** rooted blossoming tree whose closure is M endowed with the same orientation.



Proof by induction on the number of faces +
identification of the closure edges . . .

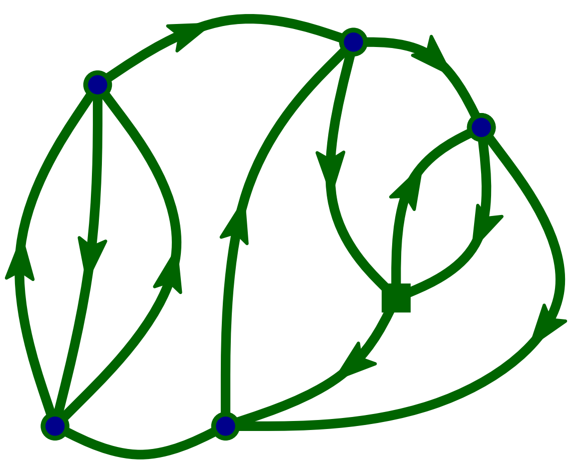
Canonical orientations

Each time a family of maps is **characterized** by the existence of some accessible orientations, we can apply this construction.

Canonical orientations

Each time a family of maps is **characterized** by the existence of some accessible orientations, we can apply this construction.

4-regular maps

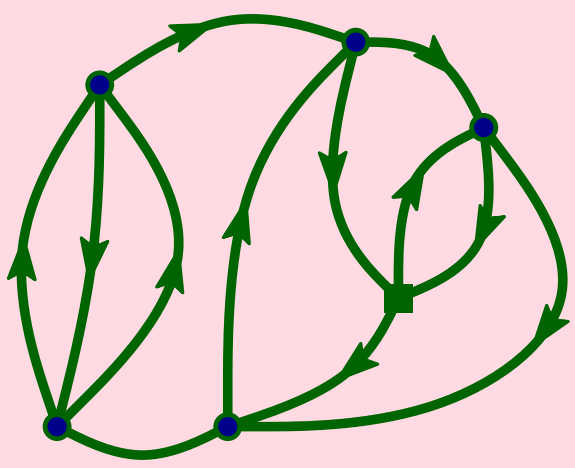


2 outgoing edges / vertex
2 ingoing edges / vertex

Canonical orientations

Each time a family of maps is **characterized** by the existence of some accessible orientations, we can apply this construction.

4-regular maps



2 outgoing edges / vertex
2 ingoing edges / vertex

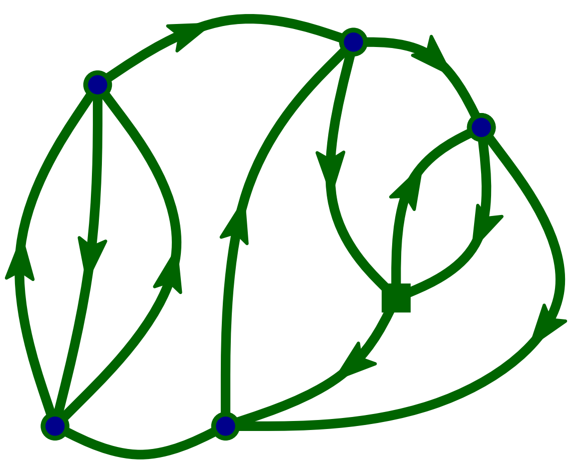
A map is 4-regular iff it admits an orientation such that each vertex has outdegree 2 and indegree 2.

Apply the general bijection to recover the result of [Schaeffer '97]

Canonical orientations

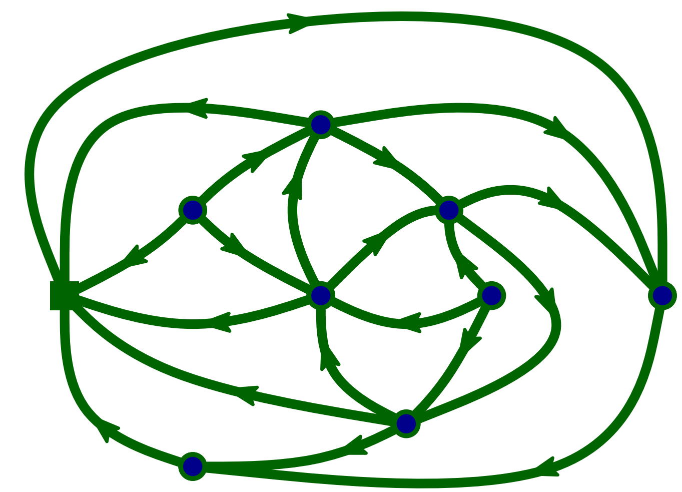
Each time a family of maps is **characterized** by the existence of some accessible orientations, we can apply this construction.

4-regular maps



2 outgoing edges / vertex
2 ingoing edges / vertex

Simple triangulations

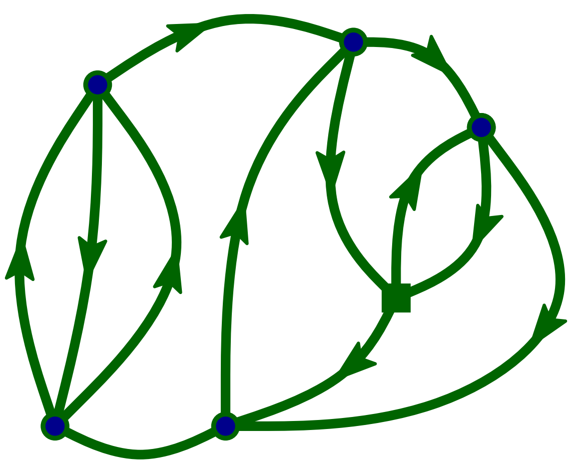


3 outgoing edges / inner vertex
1 outgoing edge / outer vertex

Canonical orientations

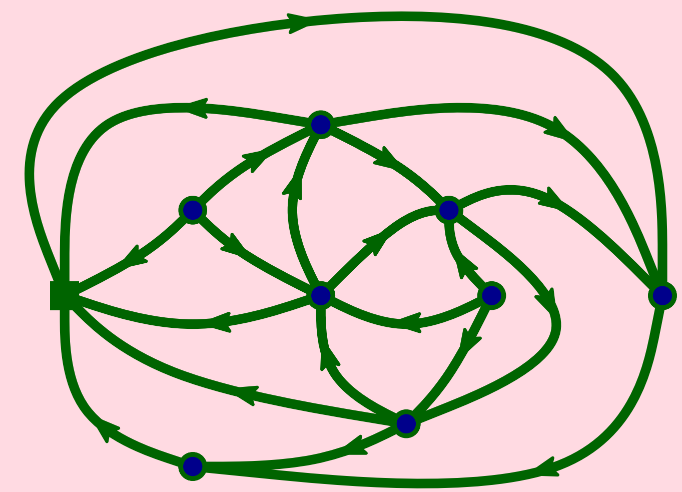
Each time a family of maps is **characterized** by the existence of some accessible orientations, we can apply this construction.

4-regular maps



2 outgoing edges / vertex
2 ingoing edges / vertex

Simple triangulations



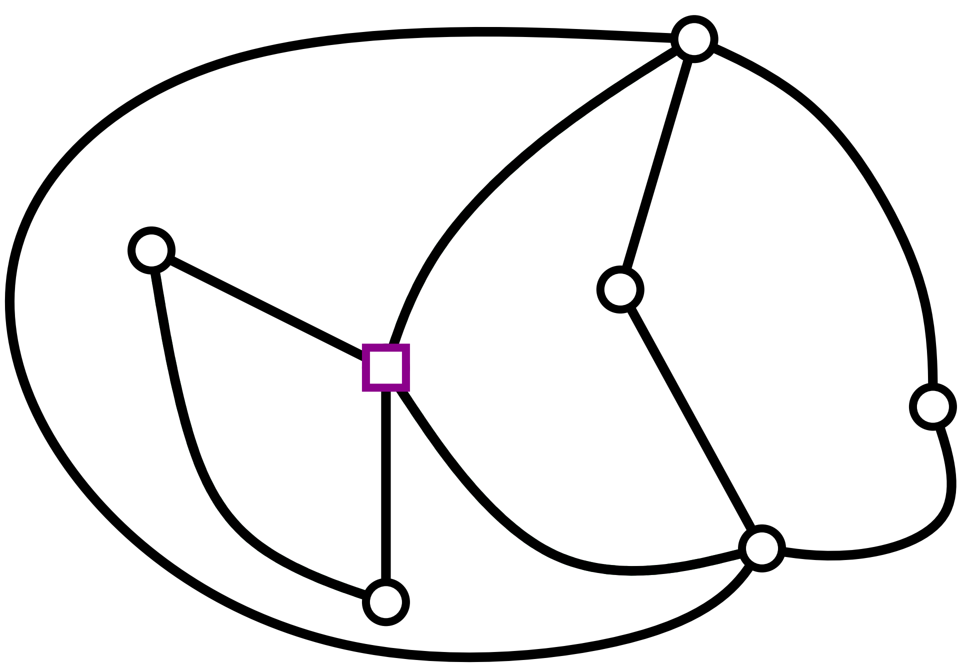
3 outgoing edges / inner vertex
1 outgoing edge / outer vertex

A triangulation is simple iff it admits an orientation such that :
each inner vertex has outdegree 3
each outer vertex has outdegree 1.

General bijection gives the result of [Poulalhon, Schaeffer '05]

Bijective method

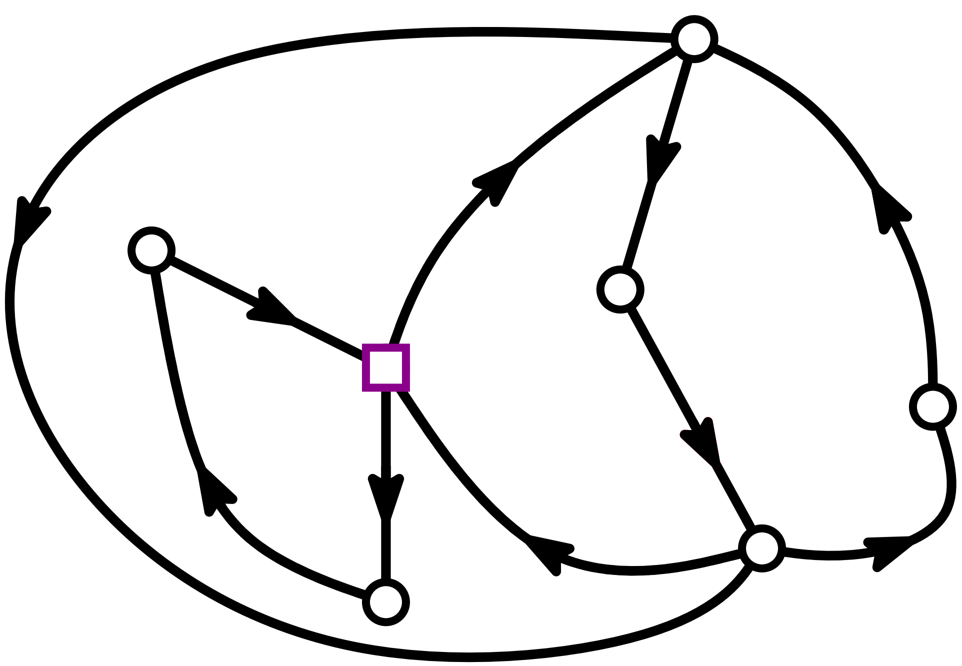
- Select a family of maps



Maps with even degree = Eulerian maps

Bijective method

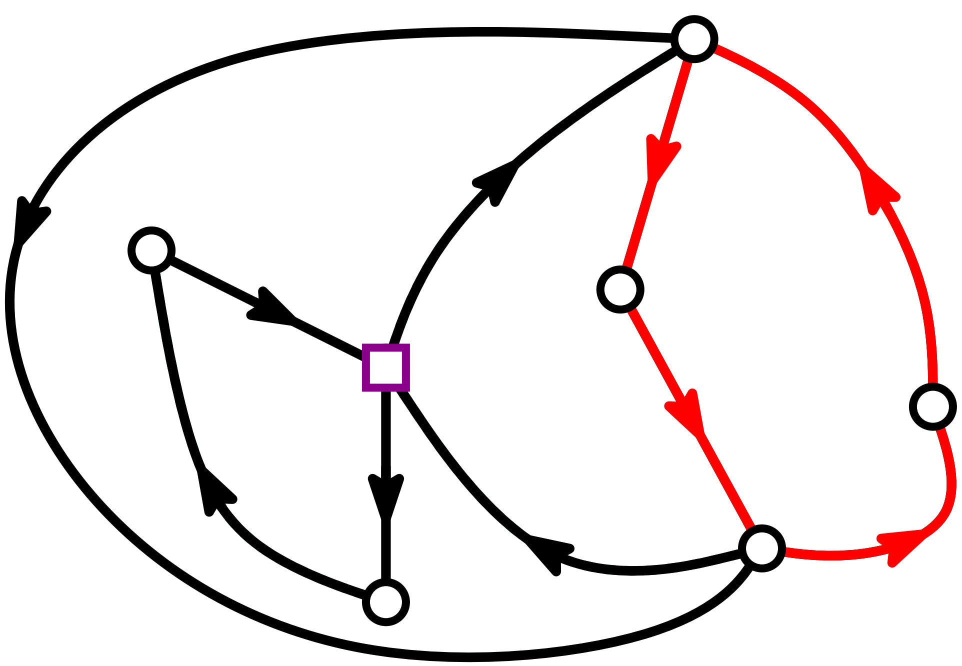
- Select a family of maps
- Find a characterization by orientations



Maps with even degree = Eulerian maps
Same in/outdegree

Bijective method

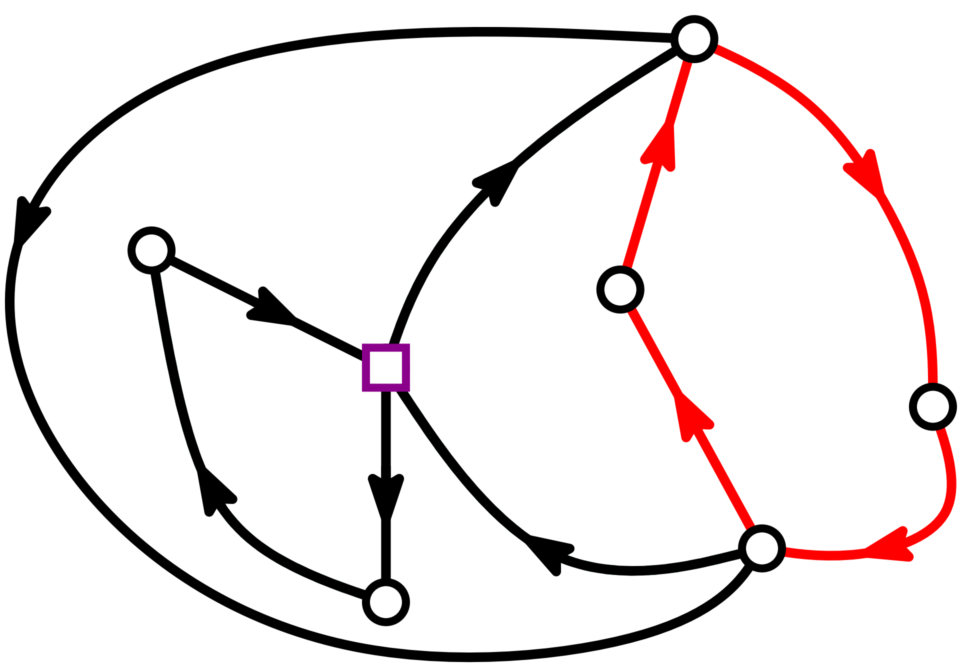
- Select a family of maps
- Find a characterization by orientations
- Consider the unique orientation without ccw cycles.



Maps with even degree = Eulerian maps
Same in/outdegree

Bijective method

- Select a family of maps
- Find a characterization by orientations
- Consider the unique orientation without ccw cycles.

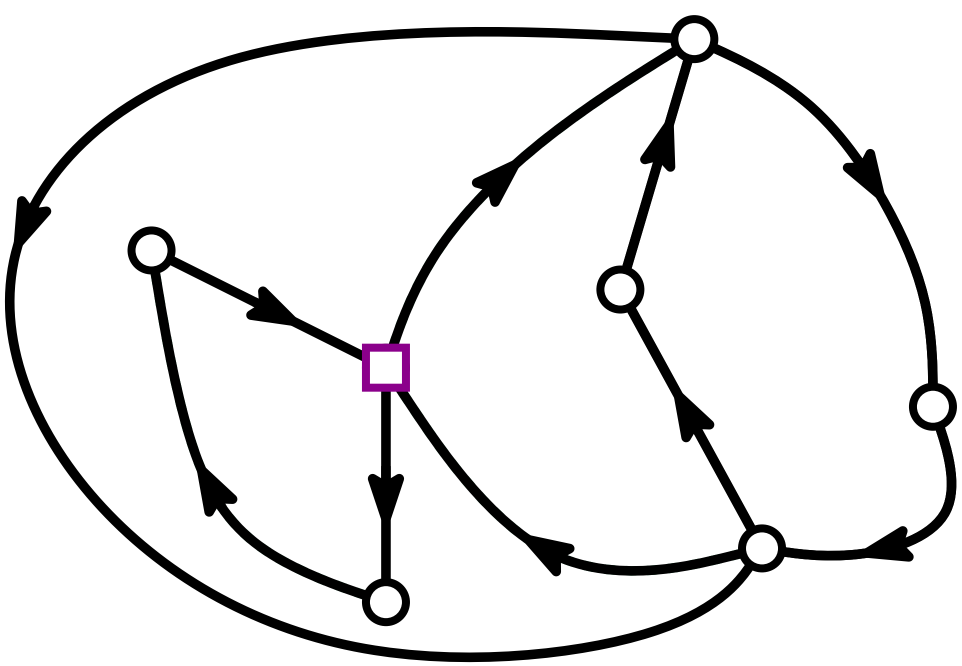


Maps with even degree = Eulerian maps

Same in/outdegree

Bijective method

- Select a family of maps
- Find a characterization by orientations
- Consider the unique orientation without ccw cycles.

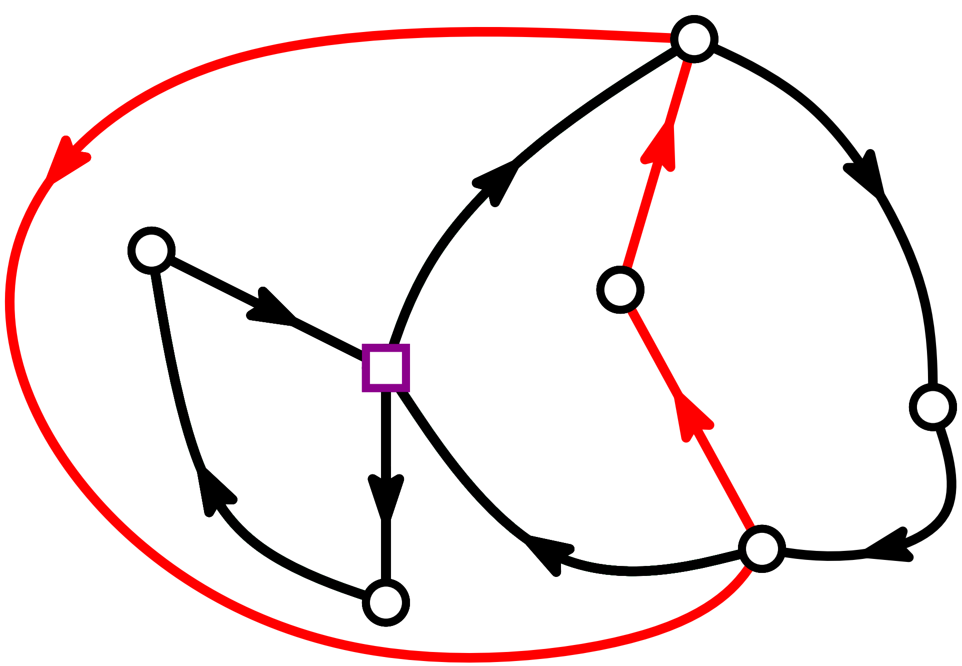


Maps with even degree = Eulerian maps

Same in/outdegree

Bijective method

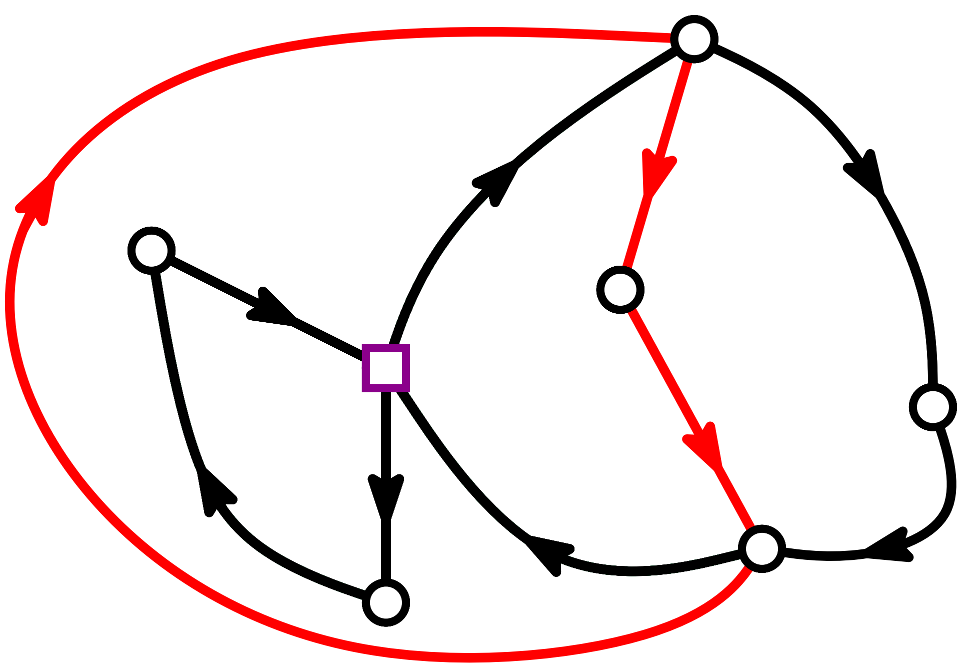
- Select a family of maps
- Find a characterization by orientations
- Consider the unique orientation without ccw cycles.



Maps with even degree = Eulerian maps
Same in/outdegree

Bijective method

- Select a family of maps
- Find a characterization by orientations
- Consider the unique orientation without ccw cycles.

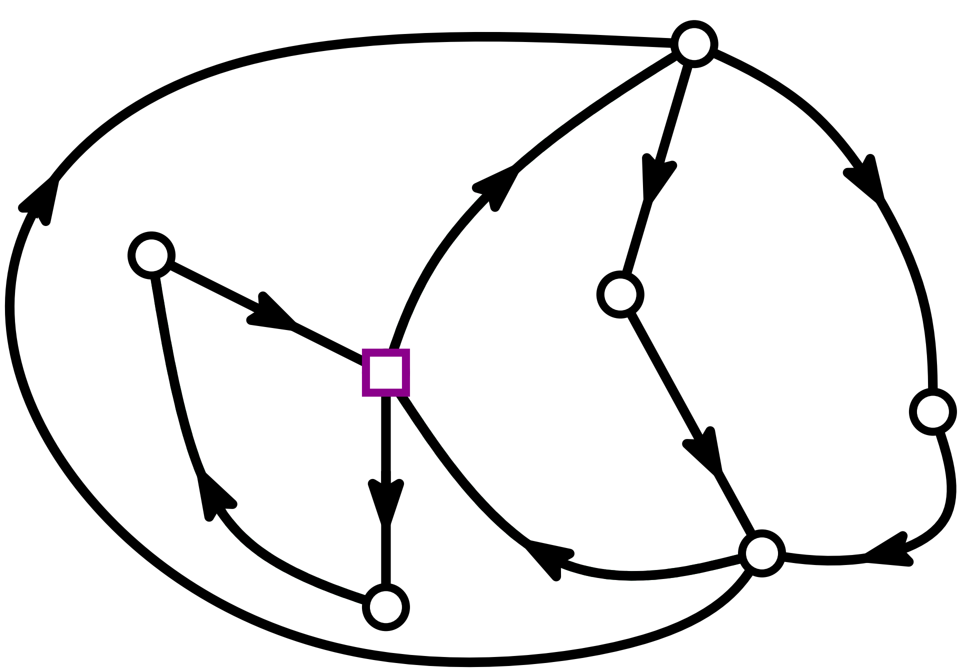


Maps with even degree = Eulerian maps

Same in/outdegree

Bijective method

- Select a family of maps
- Find a characterization by orientations
- Consider the unique orientation without ccw cycles.

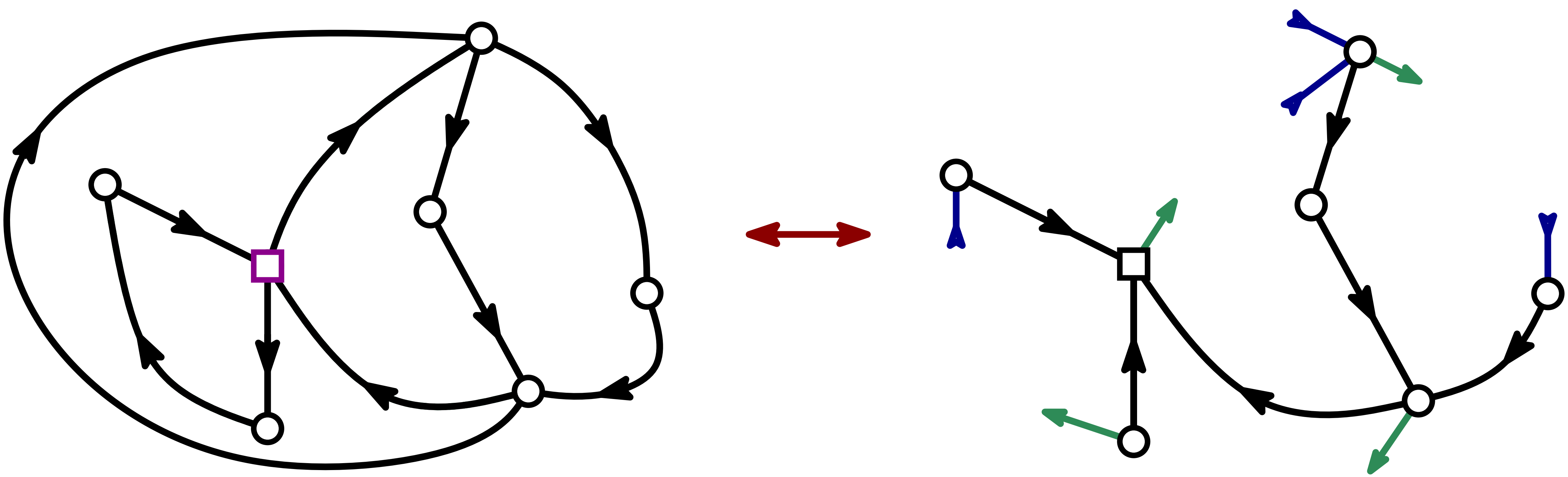


Maps with even degree = Eulerian maps

Same in/outdegree

Bijective method

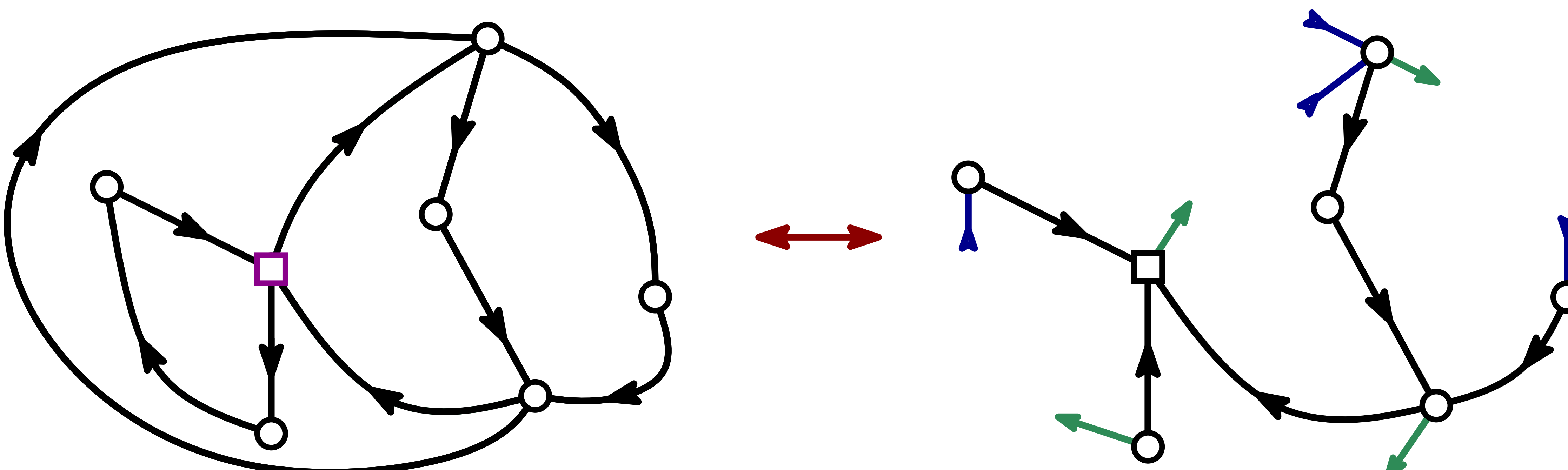
- Select a family of maps
- Find a characterization by orientations
- Consider the unique orientation without ccw cycles.
- Apply the bijection.



Maps with even degree = Eulerian maps
Same in/outdegree

Bijective method

- Select a family of maps
- Find a characterization by orientations
- Consider the unique orientation without ccw cycles.
- Apply the bijection.
- Study the family of blossoming trees.



Maps with even degree = Eulerian maps
Same in/outdegree

↔

Trees with
same indegree/outdegree.

References

Reference on the theory of α-orientations [Felsner '04] (and also [Propp '93]).

Application to straight-line drawing :

[Schnyder '89]

[Bonichon, Felsner, Mosbah '04] : refinement on Schnyder initial idea.

[Fusy's PhD '07]

Spanning trees and couplings :

[Propp '93]

[Kenyon, Propp, Wilson '00]

Thank you !

Bijections and orientations :

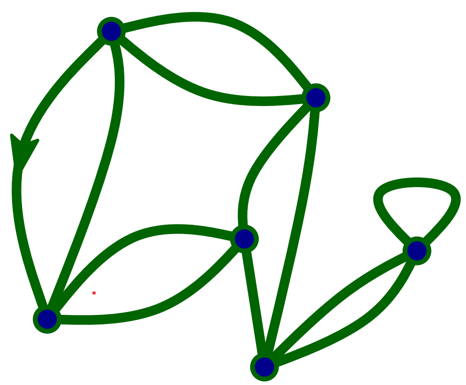
[Bernardi '07] + [A.,Poulalhon '15]

[Bernardi, Fusy '12] : unification of existing bijections relying on orientations.

[Addario-Berry, A. +14] + [Bernardi, Collet, Fusy '14] : tracking of distances.

Exercise

1) Prove (using bijections with blossoming trees) that the number of rooted 4-regular maps with n vertices is :



$$R_n = \frac{2}{n+2}\frac{3^n}{n+1}\binom{2n}{n}$$

[Tutte, 62], [Schaeffer '97]