Simple maps (also) converge to the Brownian map

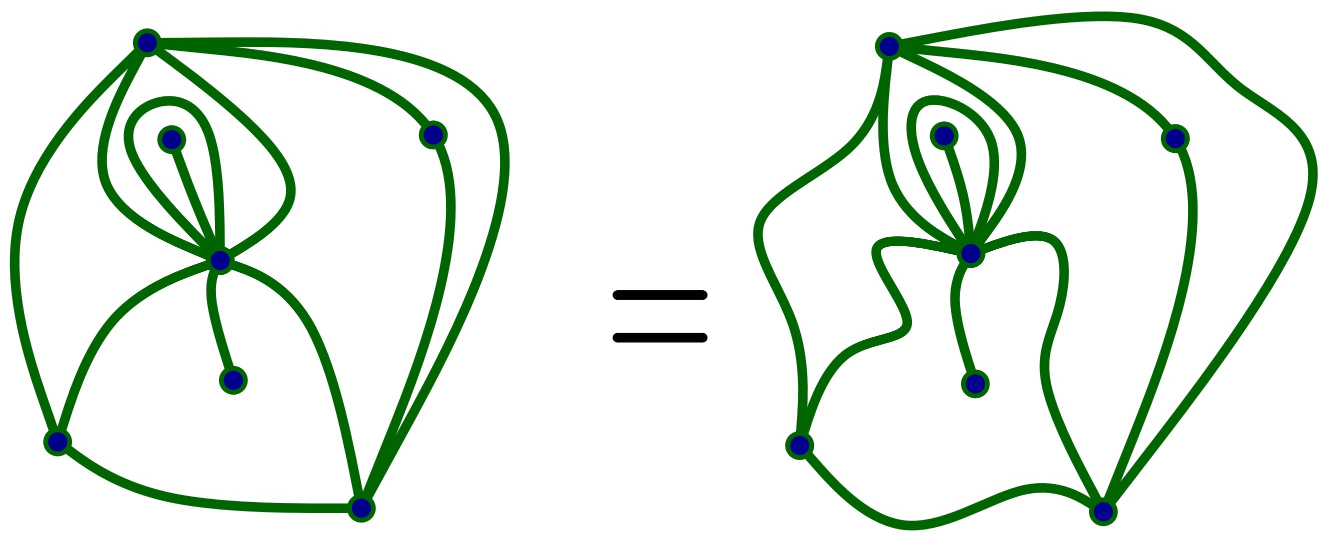
Marie Albenque (CNRS, LIX, École Polytechnique)

joint work with Olivier Bernardi (Brandeis University),
Gwendal Collet and Eric Fusy (LIX)

MAC2 Workshop,
7th July 2014

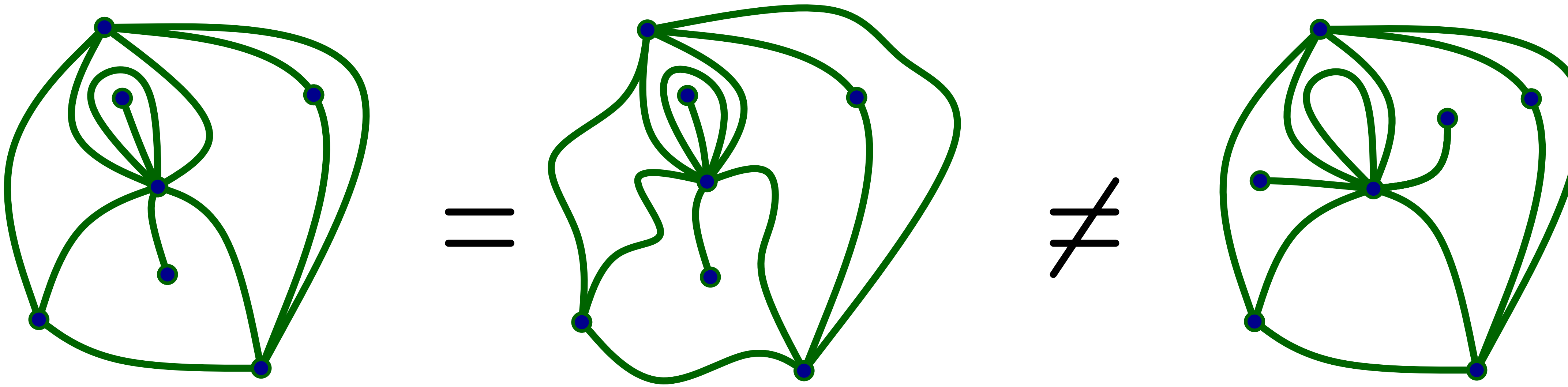
Planar Maps – Definition.

A **planar map** is the proper embedding of a planar connected graph in the 2-dimensional sphere seen up to continuous deformations.



Planar Maps – Definition.

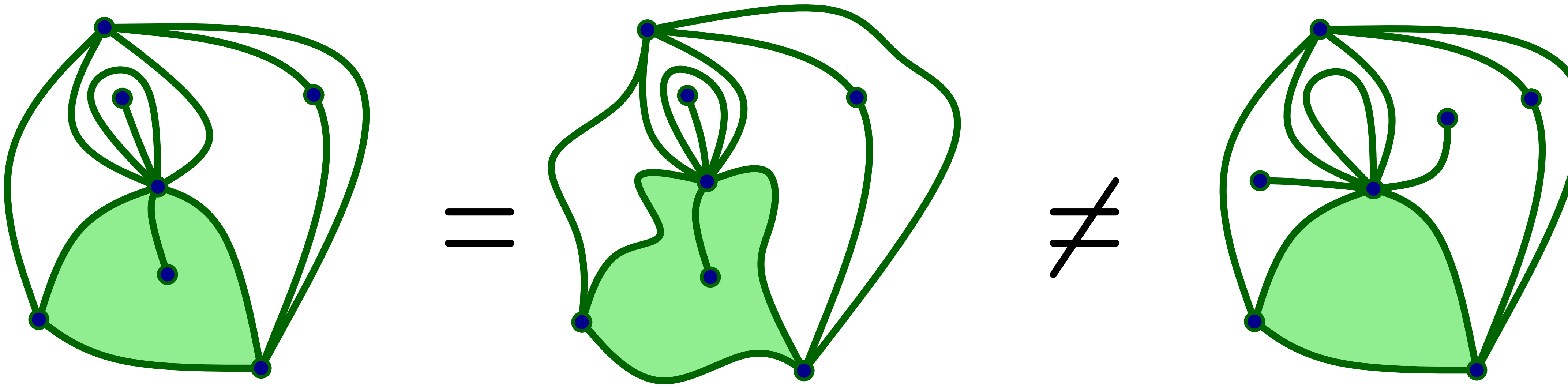
A **planar map** is the proper embedding of a planar connected graph in the 2-dimensional sphere seen up to continuous deformations.



planar map = planar graph + cyclic order of neigbours around each vertex.

Planar Maps – Definition.

A **planar map** is the proper embedding of a planar connected graph in the 2-dimensional sphere seen up to continuous deformations.

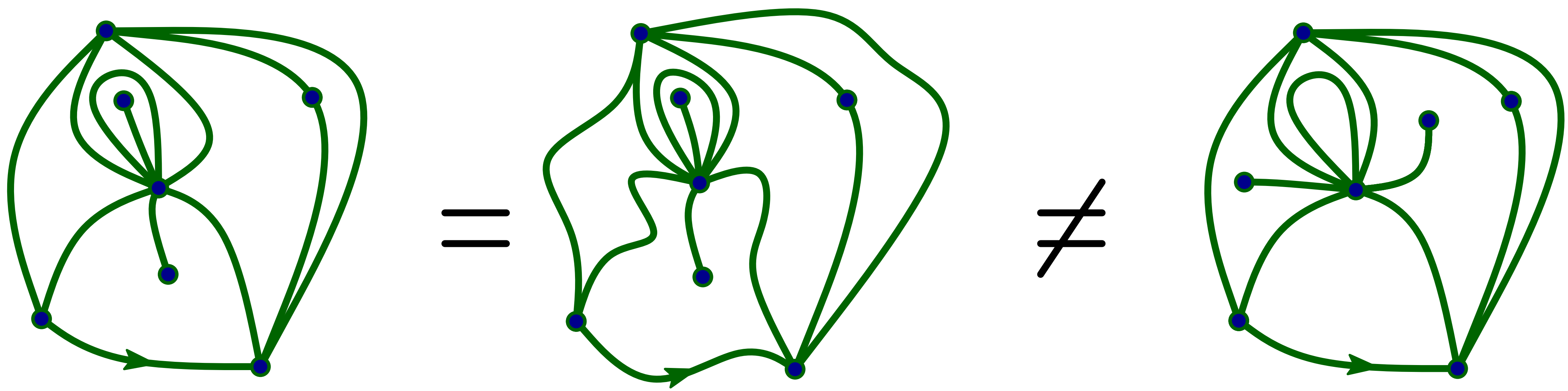


planar map = planar graph + cyclic order of neigbours around each vertex.

face = connected component of the sphere when the edges are removed

Planar Maps – Definition.

A **planar map** is the proper embedding of a planar connected graph in the 2-dimensional sphere seen up to continuous deformations.



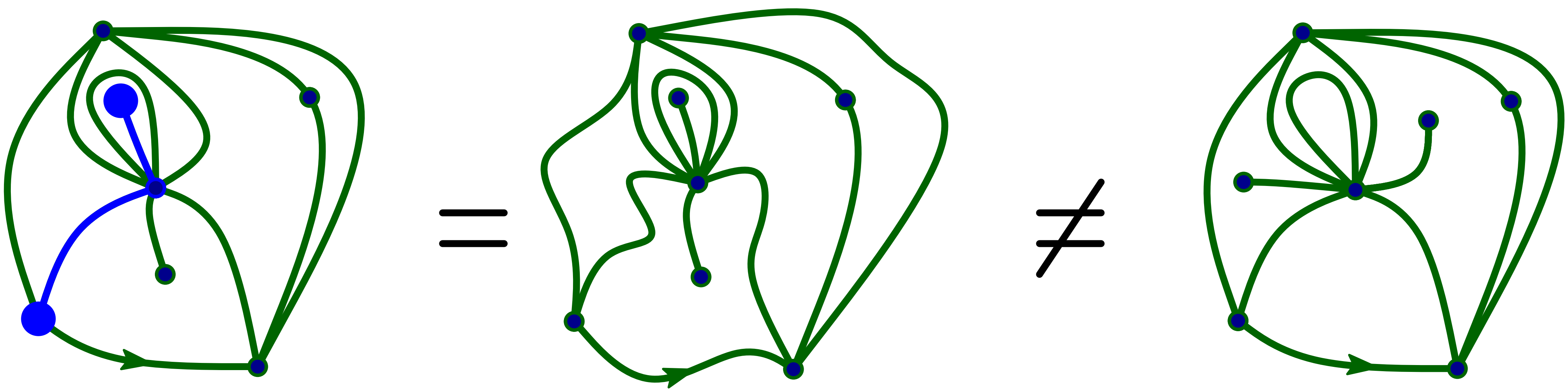
planar map = planar graph + cyclic order of neigbours around each vertex.

face = connected component of the sphere when the edges are removed

Plane maps are **rooted** : by orienting an edge.

Planar Maps – Definition.

A **planar map** is the proper embedding of a planar connected graph in the 2-dimensional sphere seen up to continuous deformations.



planar map = planar graph + cyclic order of neigbours around each vertex.

face = connected component of the sphere when the edges are removed

Plane maps are **rooted** : by orienting an edge.

Distance between two vertices = number of edges between them.
Planar map = **Metric space**

Why maps ?

What the motivation for studying maps instead of graphs ?

Because maps have **more structure** than graphs,
they are actually simpler to study.

Why maps ?

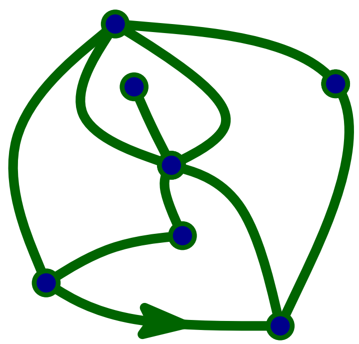
What the motivation for studying maps instead of graphs ?

Because maps have **more structure** than graphs,
they are actually simpler to study.

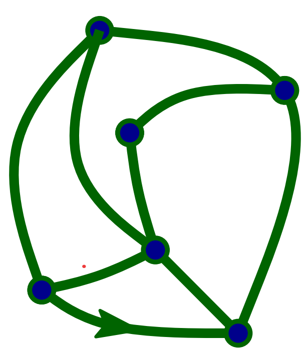
Euler Formula : $\#\,\text{vertices} + \#\,\text{faces} = \mathbf{2} + \#\,\text{edges}$

A quadrangulation with n faces has $2n$ edges and $n+2$ vertices.

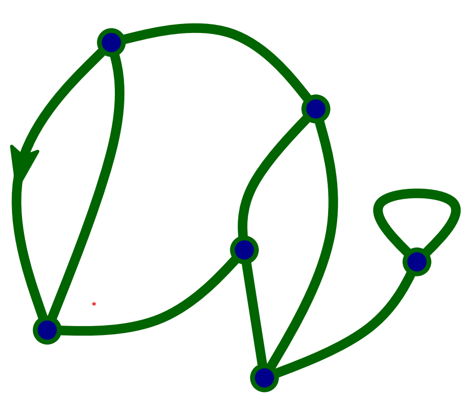
Which maps ?



Quadrangulations (all faces have degree 4)



Simple maps (no loops nor multiple edges)



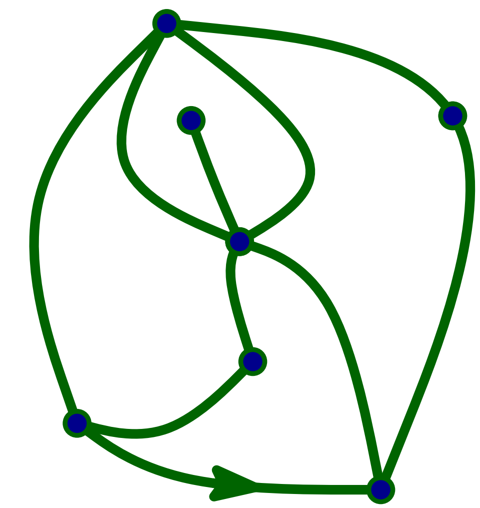
Cubic maps (all vertices have degree 3)

Random maps

$\mathcal{Q}_n = \{$Quadrangulations of size $n\}$
$= n+2$ vertices, n faces, $2n$ edges

$Q_n =$ Random element of $\mathcal{Q}_n$

$(V(Q_n), d_{gr})$ is a random compact metric space

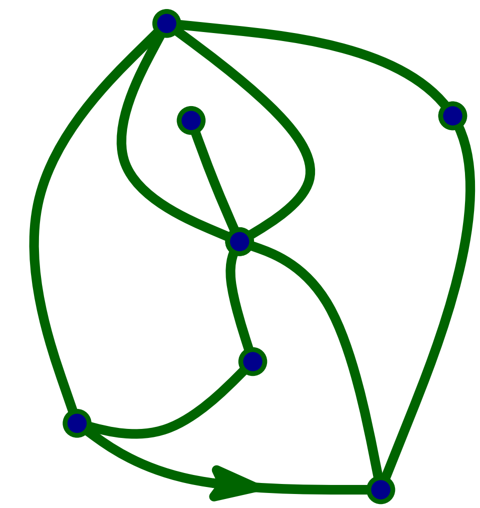


Random maps

$\mathcal{Q}_n = \{$Quadrangulations of size $n\}$
$= n + 2$ vertices, n faces, $2n$ edges

$Q_n =$ Random element of $\mathcal{Q}_n$

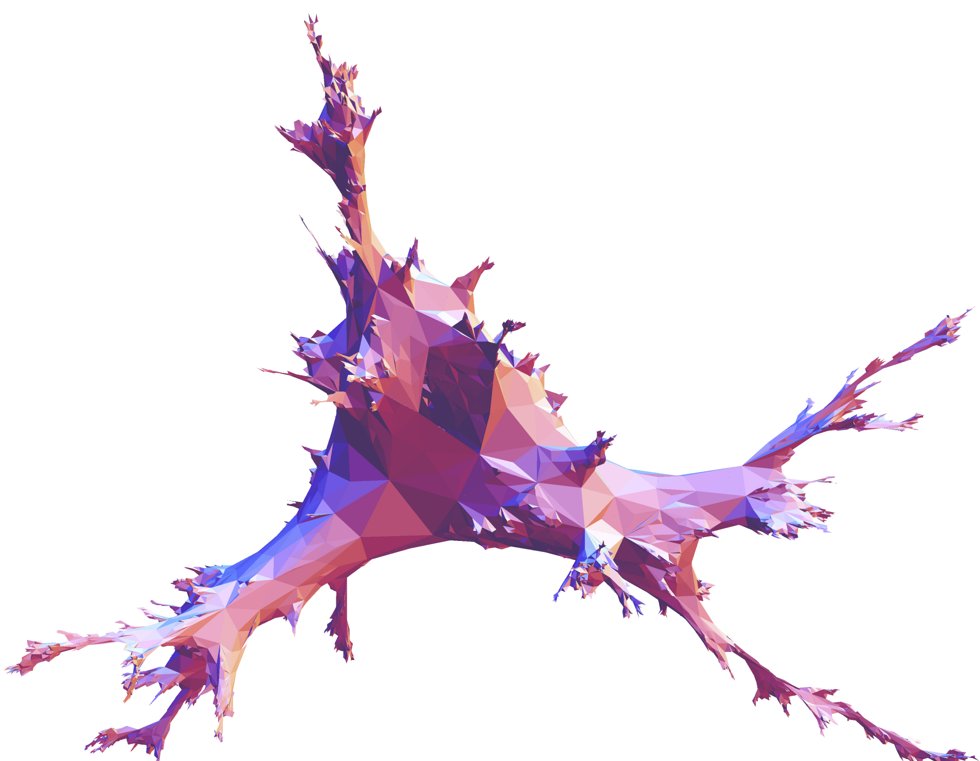
$(V(Q_n), d_{gr})$ is a random compact metric space



Simulations by N.Curien

Random maps

What is the behavior of Q_n when n goes to infinity ?
typical distances?
convergence towards a continuous object ?



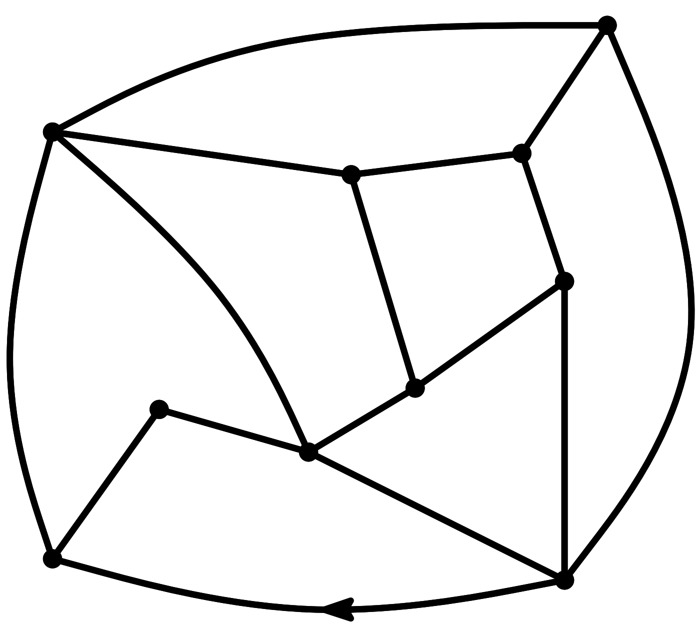
Simulations by N.Curien

Random maps

What is the behavior of Q_n when n goes to infinity ?
typical distances?
convergence towards a continuous object ?

well understood:

- Schaeffer's bijection : quadrangulations $\leftrightarrow$ labeled trees.

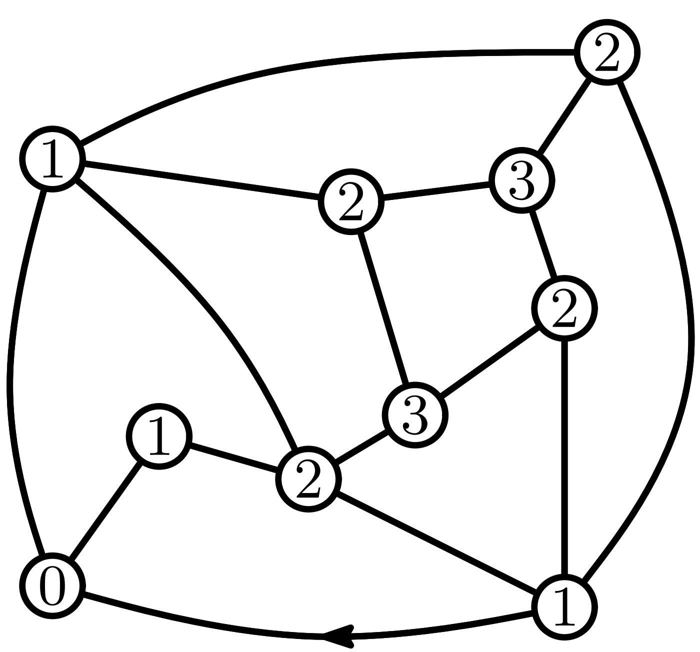


Random maps

What is the behavior of Q_n when n goes to infinity ?
typical distances?
convergence towards a continuous object ?

well understood:

- Schaeffer's bijection : quadrangulations $\leftrightarrow$ labeled trees.

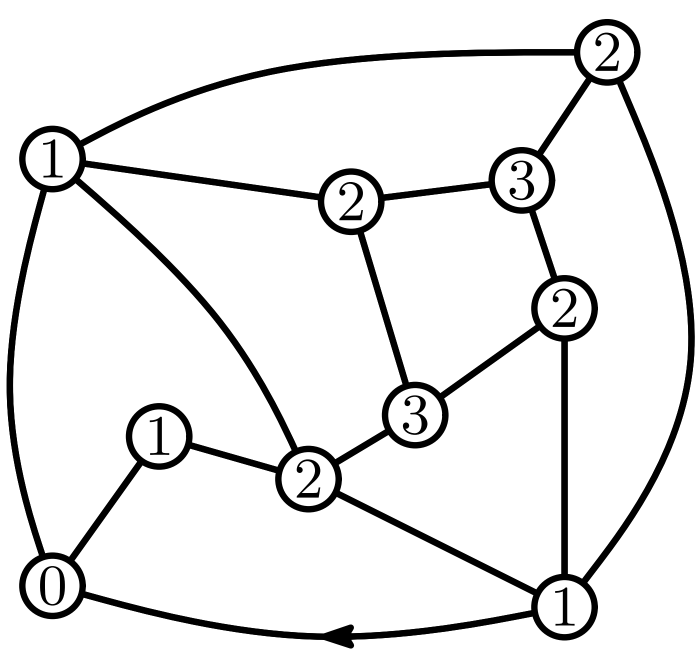


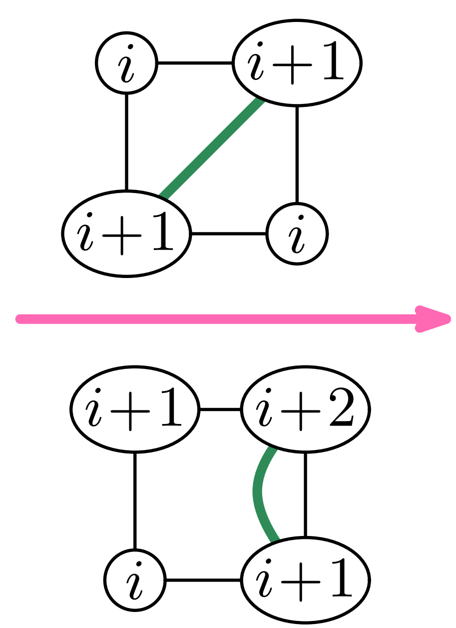
Random maps

What is the behavior of Q_n when n goes to infinity ?
typical distances?
convergence towards a continuous object ?

well understood:

- Schaeffer's bijection : quadrangulations $\leftrightarrow$ labeled trees.



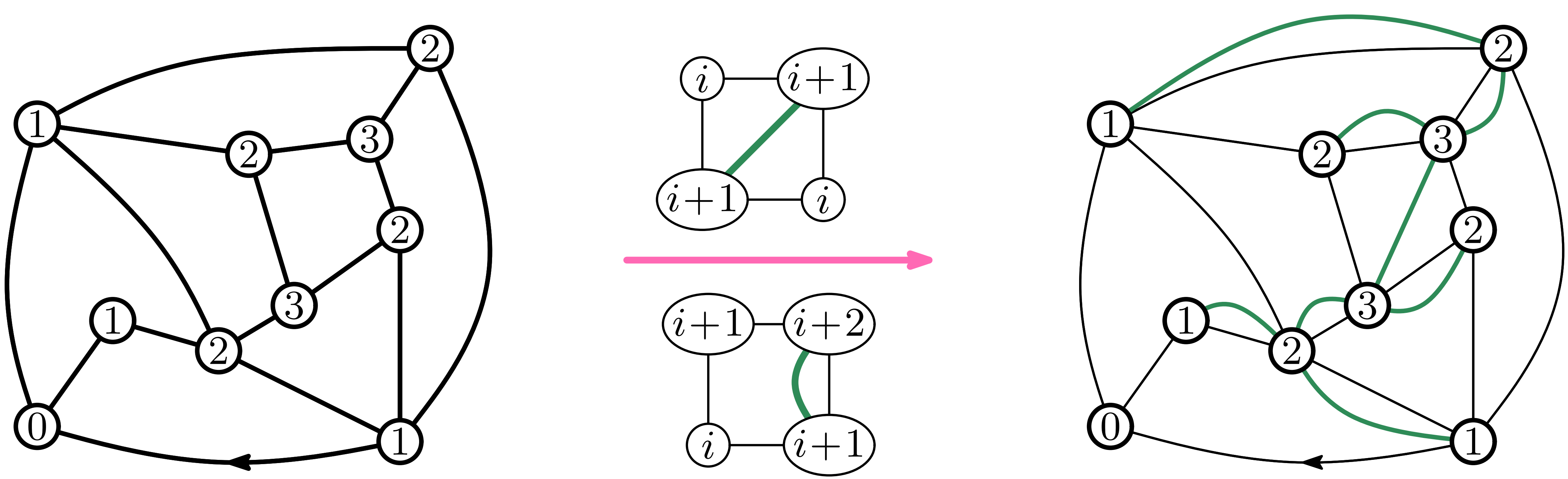


Random maps

What is the behavior of Q_n when n goes to infinity ?
typical distances?
convergence towards a continuous object ?

well understood:

- Schaeffer's bijection : quadrangulations $\leftrightarrow$ labeled trees.

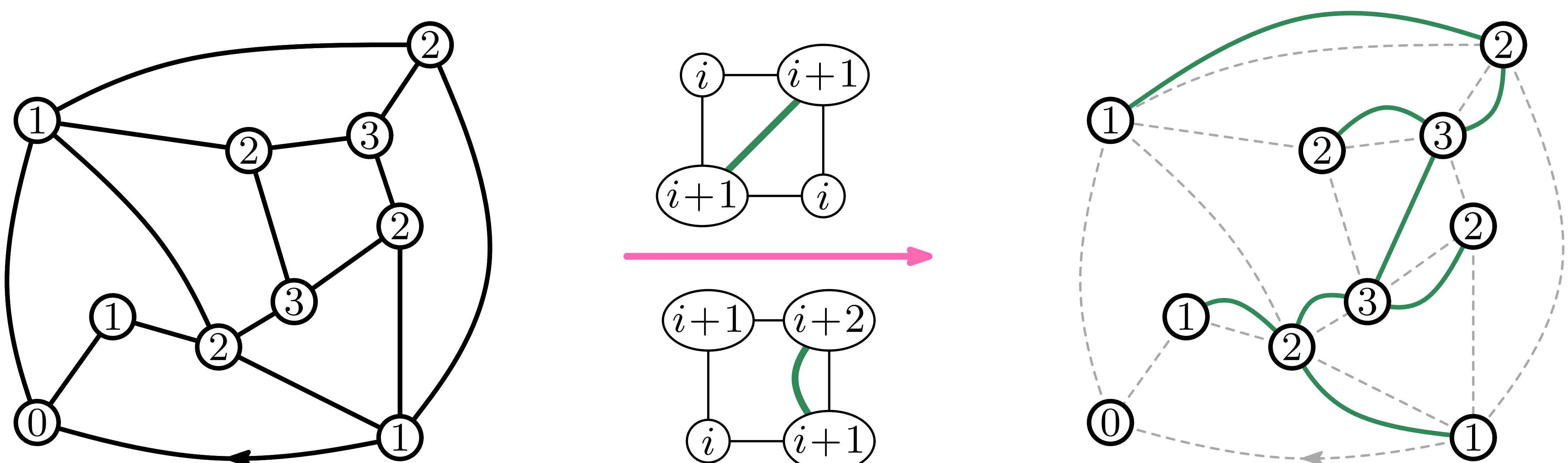


Random maps

What is the behavior of Q_n when n goes to infinity ?
typical distances?
convergence towards a continuous object ?

well understood:

• Schaeffer's bijection : quadrangulations $\leftrightarrow$ labeled trees.
Labels in the trees = distances between the vertices and the root.



Random maps

What is the behavior of Q_n when n goes to infinity ?
typical distances?
convergence towards a continuous object ?

well understood:

• Schaeffer's bijection : quadrangulations $\leftrightarrow$ labeled trees.
Labels in the trees $=$ distances between the vertices and the root.

• distance between two random points $\sim n^{1/4}$ + law of the distance
[Chassaing-Schaeffer '04]

• cvgence of normalized quadrangulations + properties of the limit
[Marckert-Mokkadem '06], [Le Gall '07], [Le Gall, Paulin '08] [Miermont '08]

Hausdorff dimension $= 4$ ([Le Gall '07])

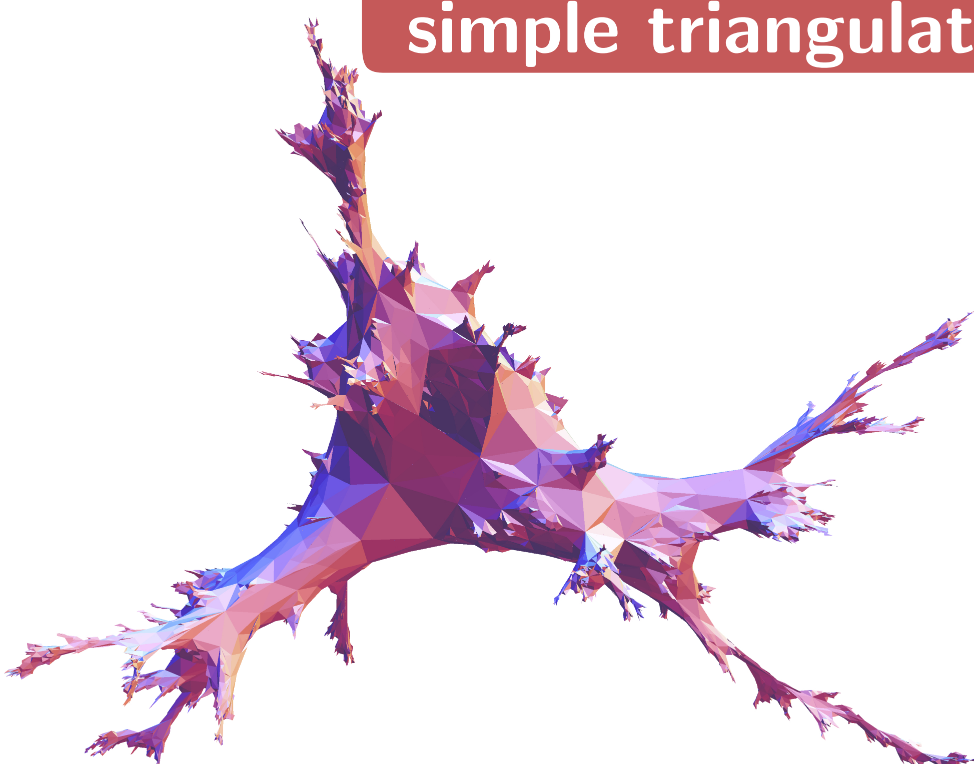
topology of the limit $=$ sphere ([Le Gall, Paulin '08], [Miermont '08])

• cvgence of normalized quadrangulations towards the **Brownian map** for Gromov-Hausdorff topology, [Miermont '13], [Le Gall '13]

Random maps

What is the behavior of Q_n when n goes to infinity ?
typical distances?
convergence towards a continuous object ?

+ what if quadrangulations are replaced by triangulations, maps, simple triangulations, simple maps, ...?



Simulations by N.Curien

Random maps

What is the behavior of Q_n when n goes to infinity ?
typical distances?
convergence towards a continuous object ?

+ what if quadrangulations are replaced by triangulations, maps, simple triangulations, simple maps, ...?

Idea : The Brownian map is a **universal** limiting object.
All "reasonable models" of maps (properly rescaled) are expected to converge towards it.

Random maps

+ what if quadrangulations are replaced by triangulations, maps, simple triangulations, simple maps, ...?

Idea: The Brownian map is a **universal** limiting object.
All "reasonable models" of maps (properly rescaled) are expected to converge towards it.

Random maps

+ what if quadrangulations are replaced by triangulations, maps, simple triangulations, simple maps, ...?

Idea: The Brownian map is a **universal** limiting object. All "reasonable models" of maps (properly rescaled) are expected to converge towards it.

So far:
- Quadrangulations [Miermont '13 + Le Gall '13]
- $2p$-angulations and triangulations [Le Gall, '13]
- Quad with no pendant vertices [Beltran, Le Gall, '13]
- Simple triangulations and quad., [Addario-Berry, A., '13]
- General maps [Betinelli, Jacob, Miermont, '13]
- Bipartite maps [Abraham, '14]

Random maps

+ what if quadrangulations are replaced by triangulations, maps, simple triangulations, simple maps, ...?

Idea: The Brownian map is a **universal** limiting object.
All "reasonable models" of maps (properly rescaled) are expected to converge towards it.

So far:
- Quadrangulations [Miermont '13 + Le Gall '13]
- $2p$-angulations and triangulations [Le Gall, '13]
- Quad with no pendant vertices [Beltran, Le Gall, '13]
- Simple triangulations and quad., [Addario-Berry, A., '13]
- General maps [Betinelli, Jacob, Miermont, '13]
- Bipartite maps [Abraham, '14]

Today:
- Simple maps [A., Bernardi, Collet, Fusy, '14]

Random maps

+ what if quadrangulations are replaced by triangulations, maps, simple triangulations, simple maps, ...?

Idea: The Brownian map is a **universal** limiting object.
All "reasonable models" of maps (properly rescaled) are expected to converge towards it.

An important remark:

Thanks to an argument of [Le Gall '13], enough to :

- understand the distance between any point and the root,
- show that the distance between two points is tight.
- prove the invariance under rerooting

and use the result of [Miermont '13], [Le Gall '13] to conclude.

So far: no **direct proof** known.

Random maps

+ what if quadrangulations are replaced by triangulations, maps, simple triangulations, simple maps, ...?

Idea: The Brownian map is a **universal** limiting object.
All "reasonable models" of maps (properly rescaled) are expected to converge towards it.

An important remark:

Thanks to an argument of [Le Gall '13], enough to :

- understand the distance between any point and the root,
- show that the distance between two points is tight.
- prove the invariance under rerooting

and use the result of [Miermont '13], [Le Gall '13] to conclude.

To do that :

- encode the maps by some trees,
- **study the limits of trees**,
- interpret the **distance in the maps by some function of the tree**.

Random maps

To do that :

- encode the maps by some trees,
- study the limits of trees,
- interpret the distance in the maps by some function of the tree.

- Quadrangulations [Miermont '13 + Le Gall '13]
- $2p$-angulations and triangulations [Le Gall, '13]
- Quad with no pendant vertices [Beltran, Le Gall, '13]
- Simple triangulations and quad. [Addario-Berry, A.,'13]
- General maps [Betinelli, Jacob, Miermont, '13]
- Bipartite maps [Abraham, '14]

Random maps

To do that :

- encode the maps by some trees,
- study the limits of trees,
- interpret the distance in the maps by some function of the tree.

with **labeled trees**
[Schaeffer '98]
[Bouttier, di Francesco, Guitter '04]

- Quadrangulations [Miermont '13 + Le Gall '13]
- $2p$-angulations and triangulations [Le Gall, '13]
- Quad with no pendant vertices [Beltran, Le Gall, '13]
- Simple triangulations and quad. [Addario-Berry, A.,'13]
- General maps [Betinelli, Jacob, Miermont, '13]
- Bipartite maps [Abraham, '14]

Random maps

To do that :

- encode the maps by some trees,
- study the limits of trees,
- interpret the distance in the maps by some function of the tree.

with **labeled trees**
[Schaeffer '98]
[Bouttier, di Francesco, Guitter '04]

with **quadrangulations**
[Ambjørn, Budd '13]

- Quadrangulations [Miermont '13 + Le Gall '13]
- $2p$-angulations and triangulations [Le Gall, '13]
- Quad with no pendant vertices [Beltran, Le Gall, '13]
- Simple triangulations and quad. [Addario-Berry, A.,'13]
- General maps [Betinelli, Jacob, Miermont, '13]
- Bipartite maps [Abraham, '14]

Random maps

To do that :

- encode the maps by some trees,
- study the limits of trees,
- interpret the distance in the maps by some function of the tree.

with **labeled trees**
[Schaeffer '98]
[Bouttier, di Francesco, Guitter '04]

with **quadrangulations**
[Ambjørn, Budd '13]

with **blossoming trees**
[Poulalhon, Schaeffer '05] [Fusy '07]

- Quadrangulations [Miermont '13 + Le Gall '13]
- $2p$-angulations and triangulations [Le Gall, '13]
- Quad with no pendant vertices [Beltran, Le Gall, '13]
- Simple triangulations and quad. [Addario-Berry, A.,'13]
- General maps [Betinelli, Jacob, Miermont, '13]
- Bipartite maps [Abraham, '14]

Random maps

To do that :

- encode the maps by some trees,
- study the limits of trees,
- interpret the distance in the maps by some function of the tree.

with **labeled trees**
[Schaeffer '98]
[Bouttier, di Francesco, Guitter '04]

with **quadrangulations**
[Ambjørn, Budd '13]

with **blossoming trees**
[Poulalhon, Schaeffer '05] [Fusy '07]
[Schaeffer,Bousquet-Mélou '00]
[Bernardi, Collet, Fusy '14]

- Quadrangulations [Miermont '13 + Le Gall '13]
- $2p$-angulations and triangulations [Le Gall, '13]
- Quad with no pendant vertices [Beltran, Le Gall, '13]
- Simple triangulations and quad. [Addario-Berry, A.,'13]
- General maps [Betinelli, Jacob, Miermont, '13]
- Bipartite maps [Abraham, '14]
- Simple maps [A., Bernardi, Collet, Fusy, '14]

More precisely: the result

Theorem : [A., Bernardi, Collet, Fusy]
$\mathcal{S}_n = \{$ simple maps with n edges $\}$
$S_n =$ uniform random element of $\mathcal{S}_n$. Then:

$$\left(V(S_n), \left(\frac{1}{2n}\right)^{1/4} d_{S_n} \right) \xrightarrow{(d)} (M, D^\star),$$

for the distance of Gromov-Hausdorff on the isometry classes of compact metric spaces.

More precisely: the result

Theorem : [A., Bernardi, Collet, Fusy]
$\mathcal{S}_n = \{$ simple maps with n edges $\}$
$S_n =$ uniform random element of $\mathcal{S}_n$. Then:

$$\left(V(S_n), \left(\frac{1}{2n}\right)^{1/4} d_{S_n} \right) \xrightarrow{(d)} (M, D^\star),$$

for the distance of Gromov-Hausdorff on the isometry classes of compact metric spaces.

- same scaling $n^{1/4}$ as for general maps

More precisely: the result

Theorem : [A., Bernardi, Collet, Fusy]
$\mathcal{S}_n = \{$ simple maps with n edges $\}$
$S_n =$ uniform random element of $\mathcal{S}_n$. Then:

$$\left(V(S_n), \left(\frac{1}{2n}\right)^{1/4} d_{S_n}\right) \xrightarrow{(d)} (M, D^\star),$$

for the distance of Gromov-Hausdorff on the isometry classes of compact metric spaces.

- same scaling $n^{1/4}$ as for general maps
- roughly : need to understand the distance between any pair of points

More precisely: the result

Theorem : [A., Bernardi, Collet, Fusy]
$\mathcal{S}_n = \{$ simple maps with n edges $\}$
$S_n =$ uniform random element of $\mathcal{S}_n$. Then:

$$\left(V(S_n), \left(\frac{1}{2n}\right)^{1/4} d_{S_n} \right) \xrightarrow{(d)} (M, D^\star),$$

for the distance of Gromov-Hausdorff on the isometry classes of compact metric spaces.

- same scaling $n^{1/4}$ as for general maps
- roughly : need to understand the distance between any pair of points
- The Brownian Map

Random maps

To do that :

- **encode the maps by some trees**,
- study the limits of trees,
- interpret the distance in the maps by some function of the tree.

with **labeled trees**
[Schaeffer '98]
[Bouttier, di Francesco, Guitter '04]

with **quadrangulations**
[Ambjørn, Budd '03]

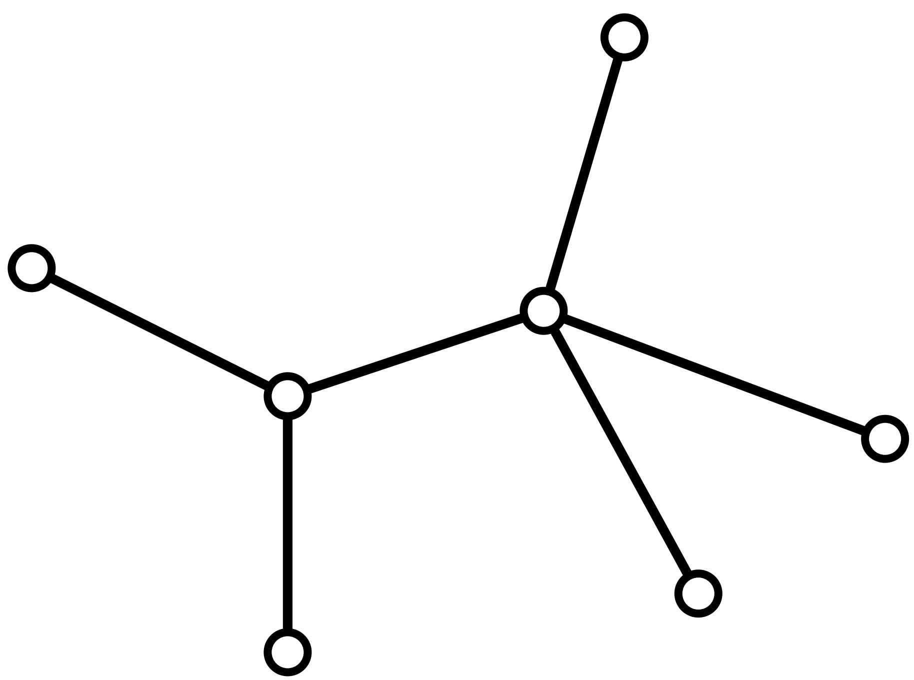
with **blossoming trees**
[Poulalhon, Schaeffer '05] [Fusy '07]
[Schaeffer,Bousquet-Mélou '00]
[Bernardi, Collet, Fusy '14]

- Quadrangulations [Miermont '13 + Le Gall '13]
- $2p$-angulations and triangulations [Le Gall, '13]
- Quad with no pendant vertices [Beltran, Le Gall, '13]
- Simple triangulations and quad. [Addario-Berry, A.,'13]
- General maps [Betinelli, Jacob, Miermont, '13]
- Bipartite maps [Abraham, '14]
- Simple maps [A., Bernardi, Collet, Fusy, '14]

What is a blossoming tree ?

A **blossoming tree** is a plane tree where vertices can carry **opening stems** or **closing stems**, such that :

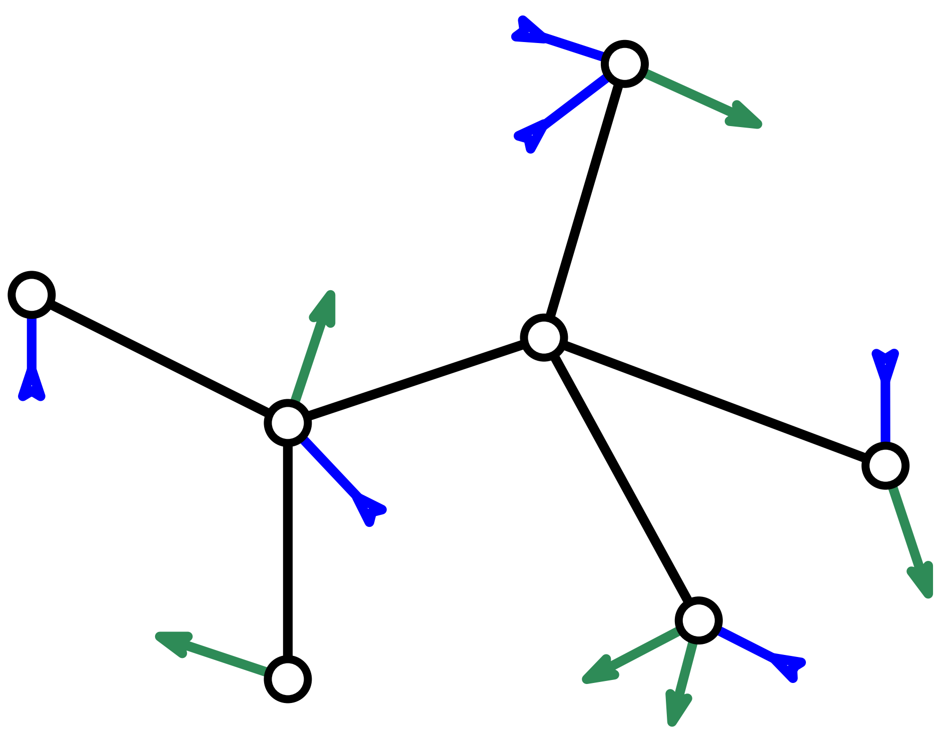
$$\# \text{ closing stems} = \# \text{ opening stems}$$



What is a blossoming tree ?

A **blossoming tree** is a plane tree where vertices can carry **opening stems** or **closing stems**, such that :

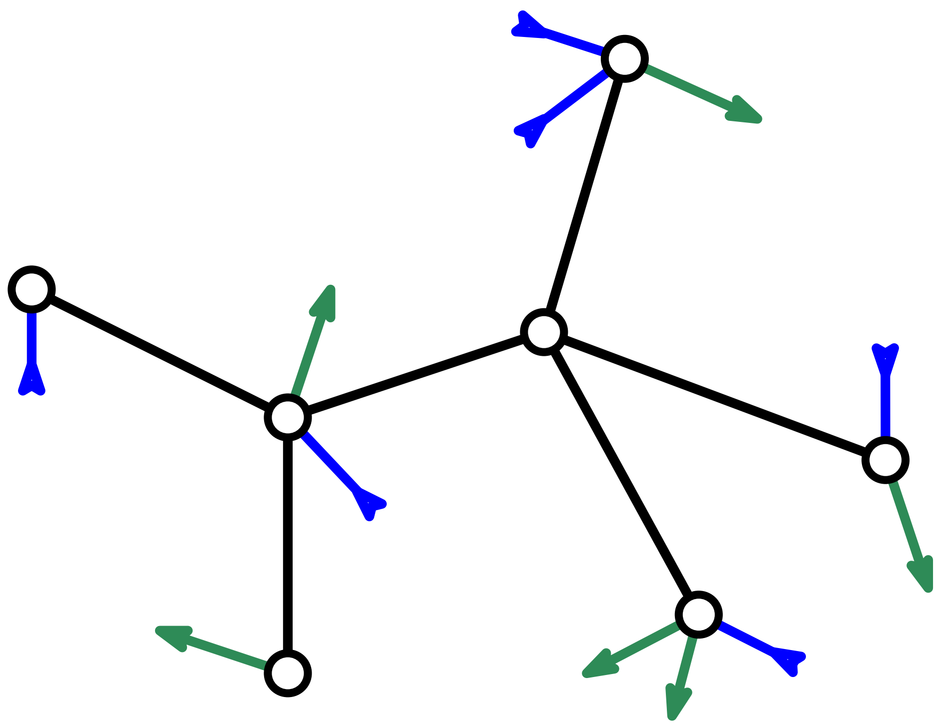
$$\# \text{ closing stems} = \# \text{ opening stems}$$

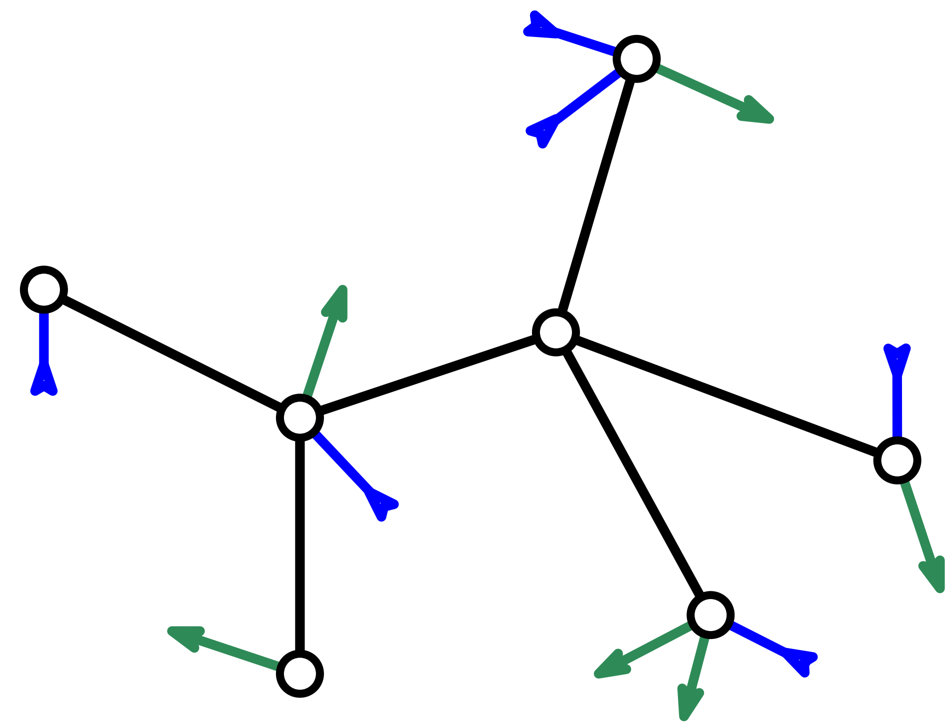


What is a blossoming tree ?

A **blossoming tree** is a plane tree where vertices can carry **opening stems** or **closing stems**, such that :

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

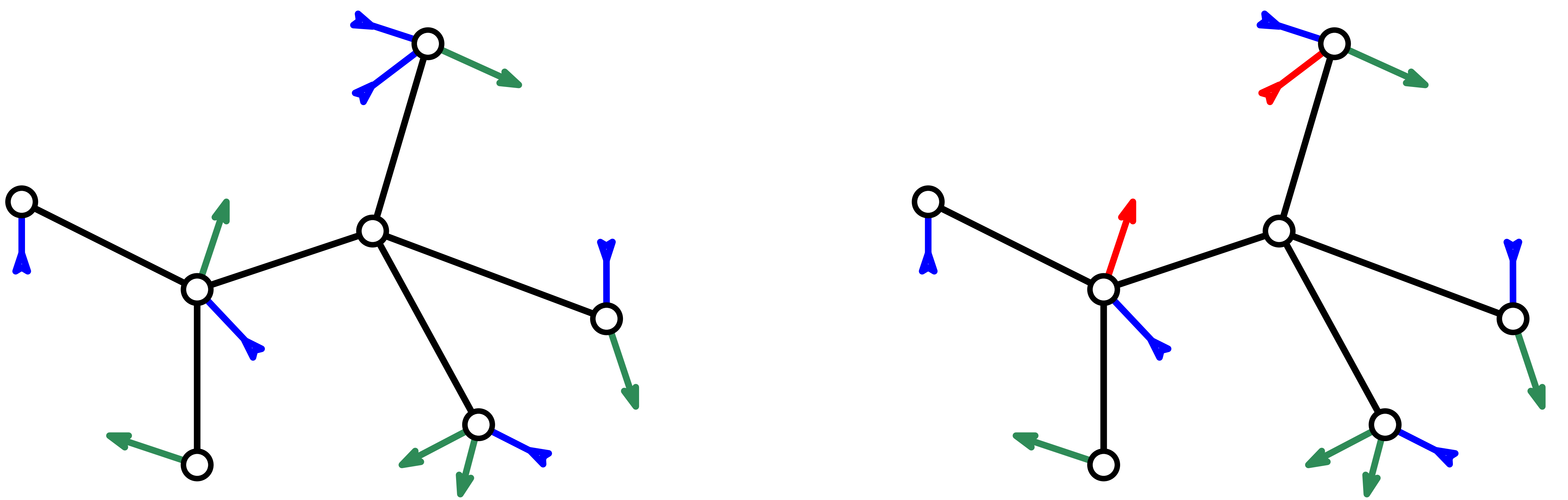




What is a blossoming tree ?

A **blossoming tree** is a plane tree where vertices can carry **opening stems** or **closing stems**, such that :

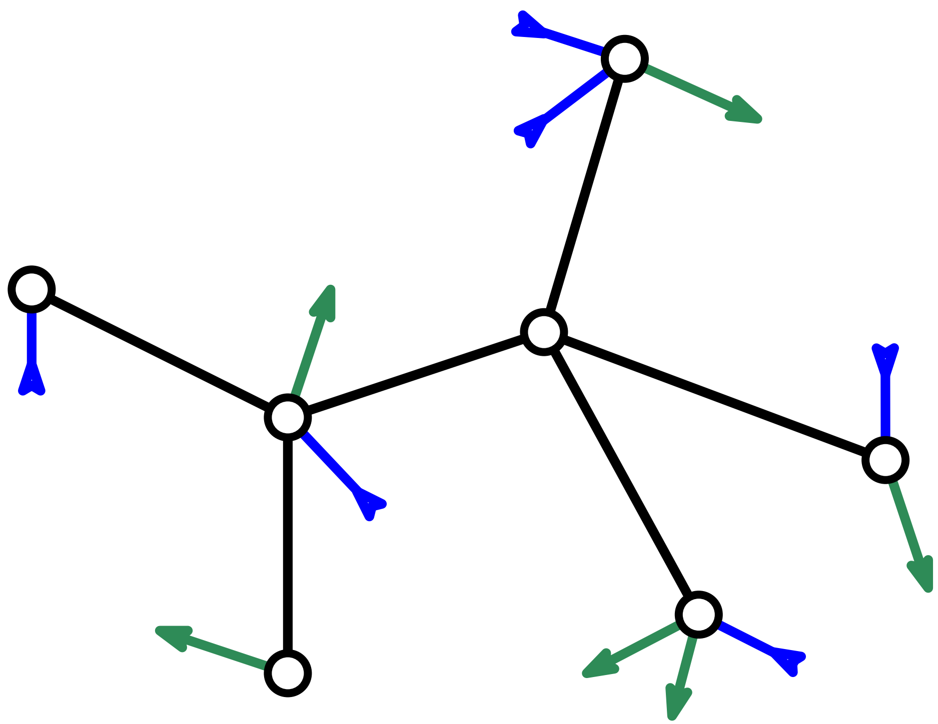
$$\# \text{ closing stems} = \# \text{ opening stems}$$

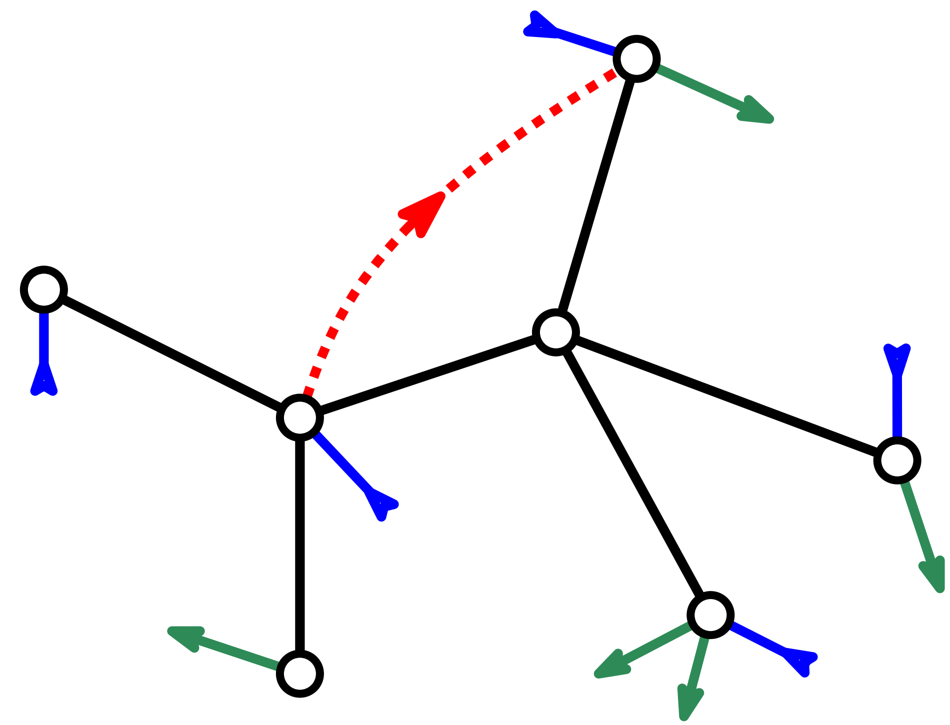


What is a blossoming tree ?

A **blossoming tree** is a plane tree where vertices can carry **opening stems** or **closing stems**, such that :

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

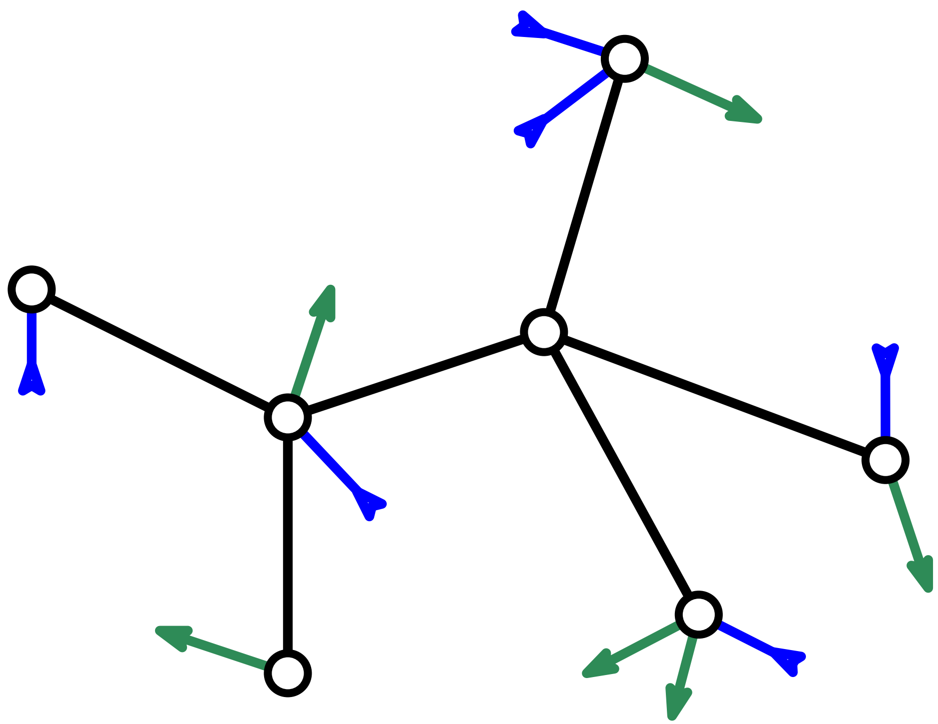


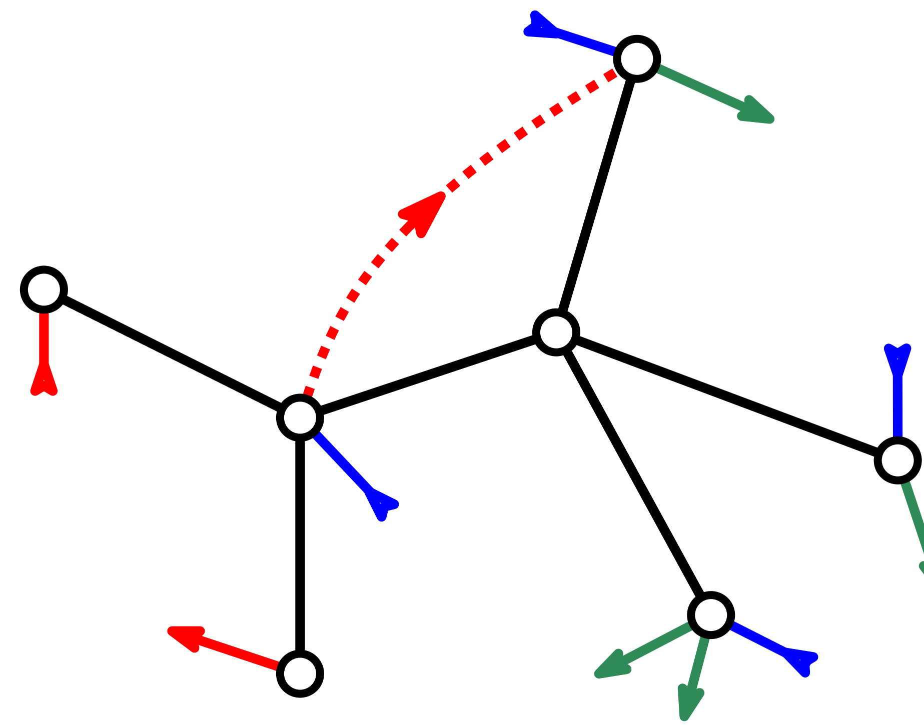


What is a blossoming tree ?

A **blossoming tree** is a plane tree where vertices can carry **opening stems** or **closing stems**, such that :

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

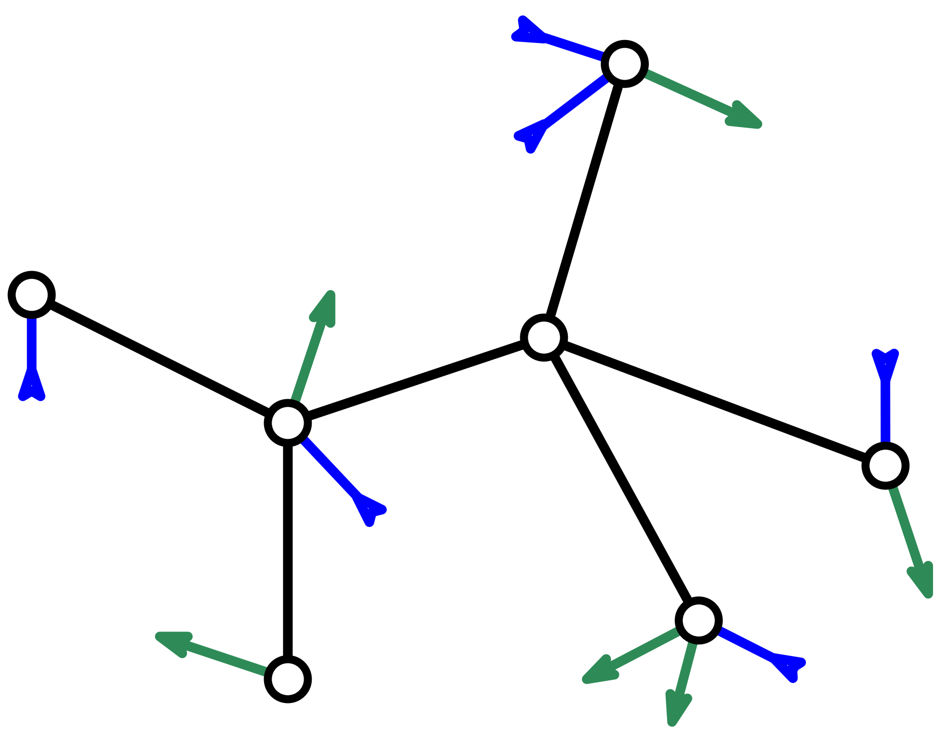


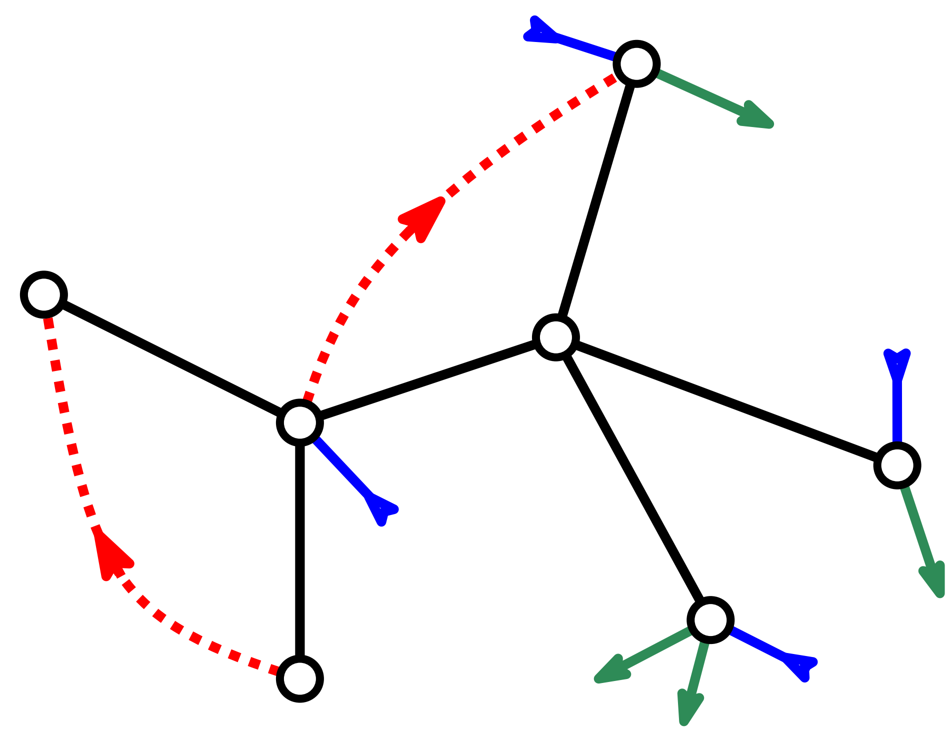


What is a blossoming tree ?

A **blossoming tree** is a plane tree where vertices can carry **opening stems** or **closing stems**, such that :

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

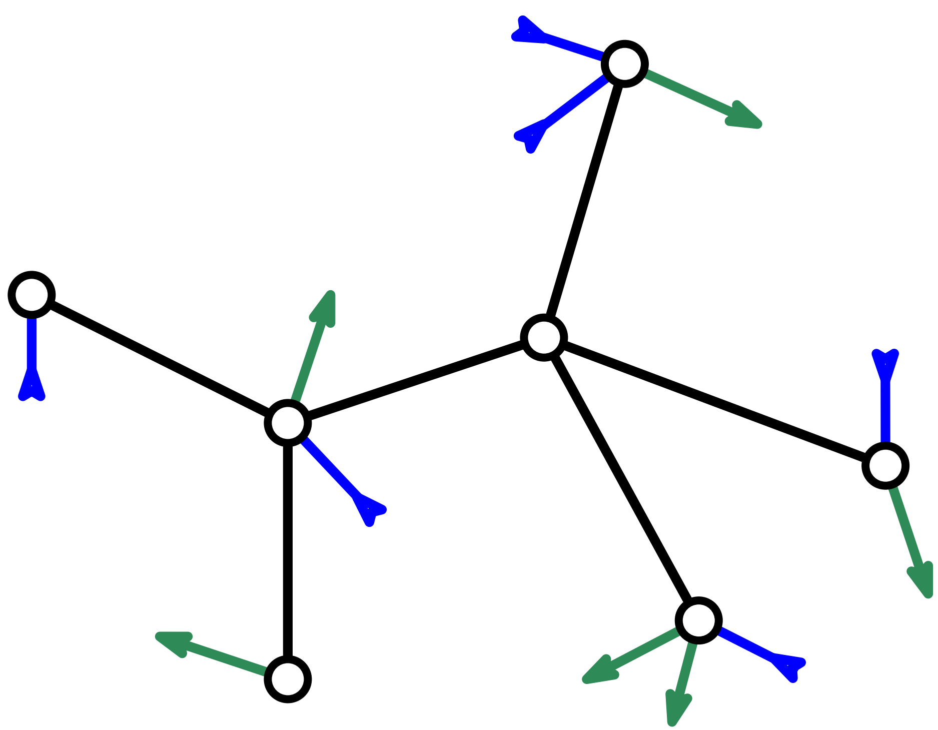


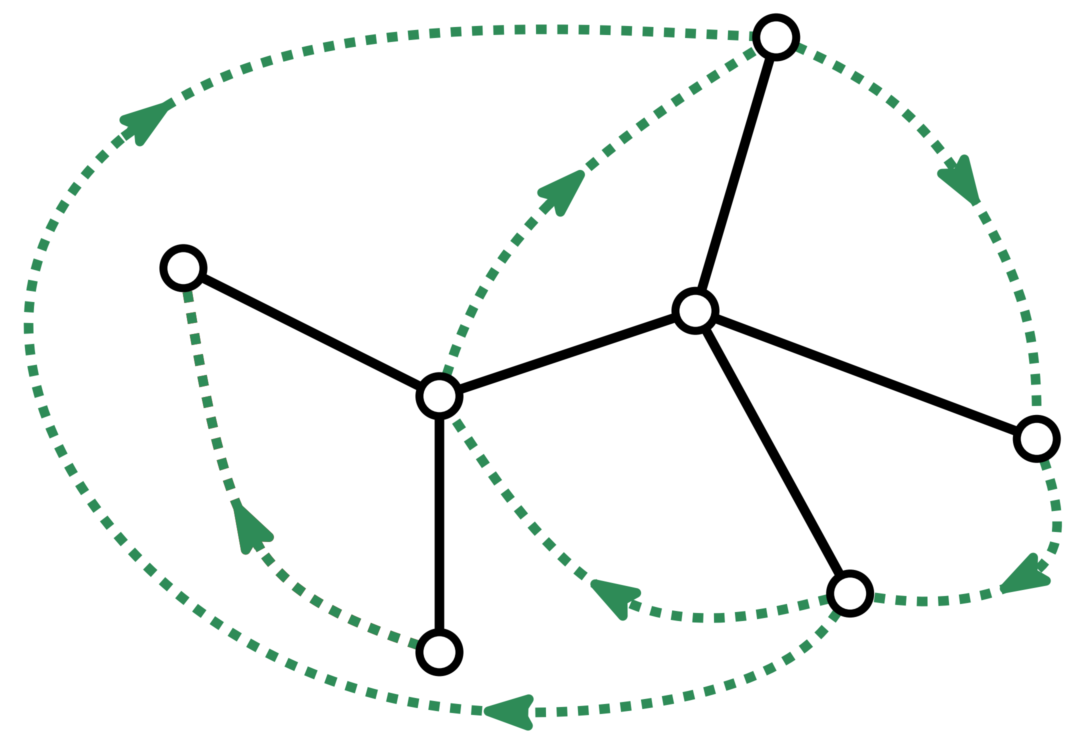


What is a blossoming tree ?

A **blossoming tree** is a plane tree where vertices can carry **opening stems** or **closing stems**, such that :

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

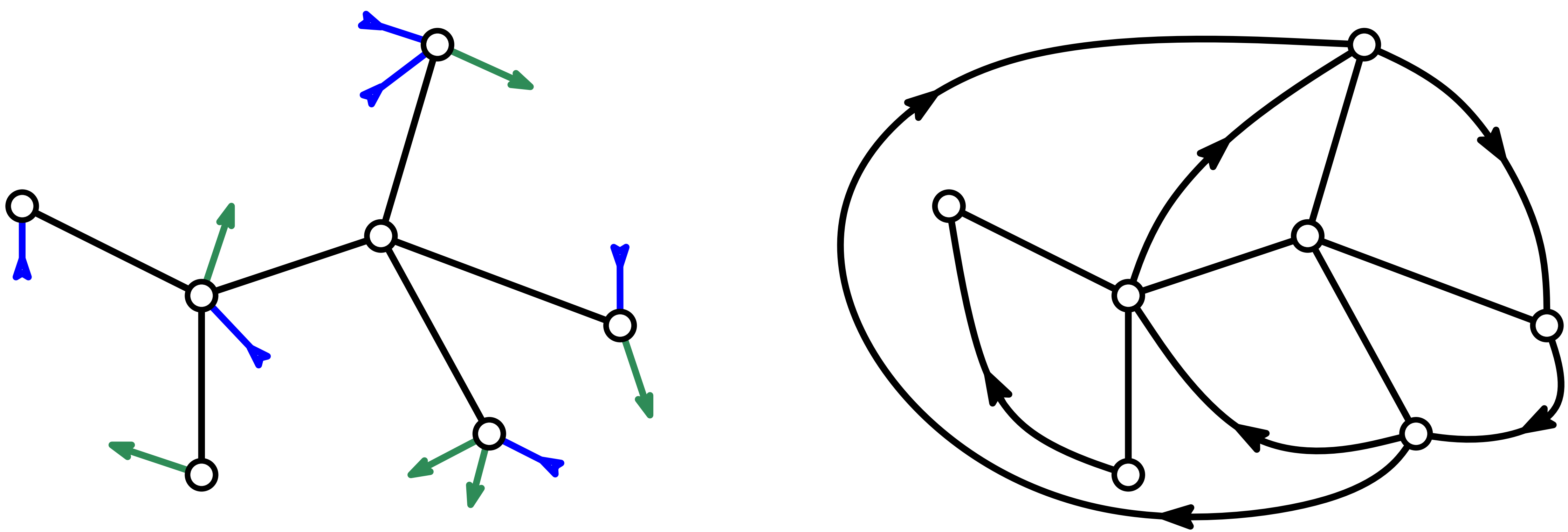




What is a blossoming tree ?

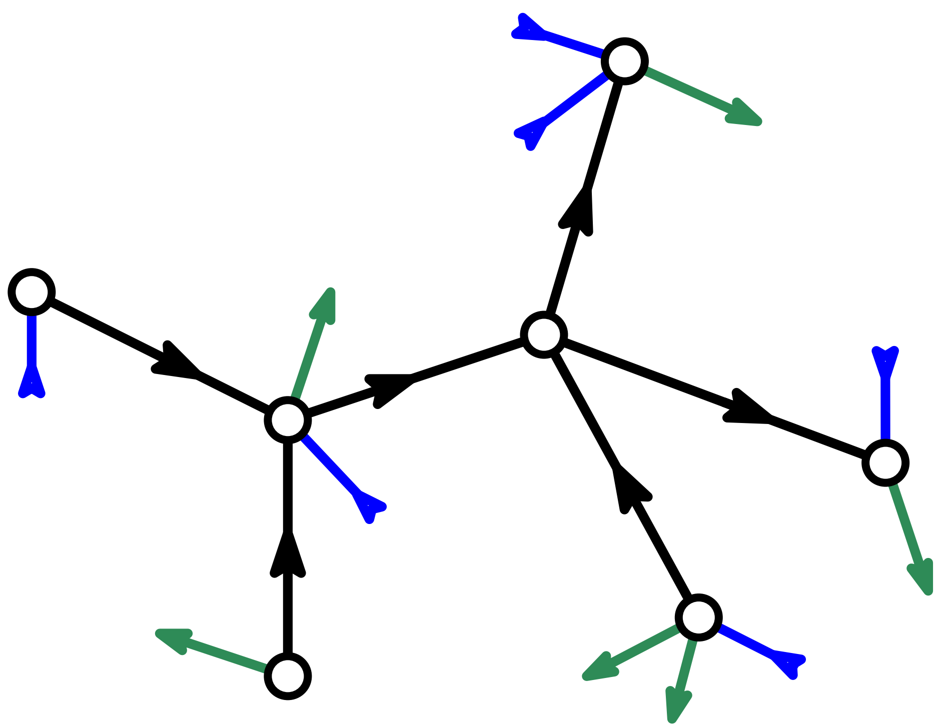
A **blossoming tree** is a plane tree where vertices can carry **opening stems** or **closing stems**, such that :

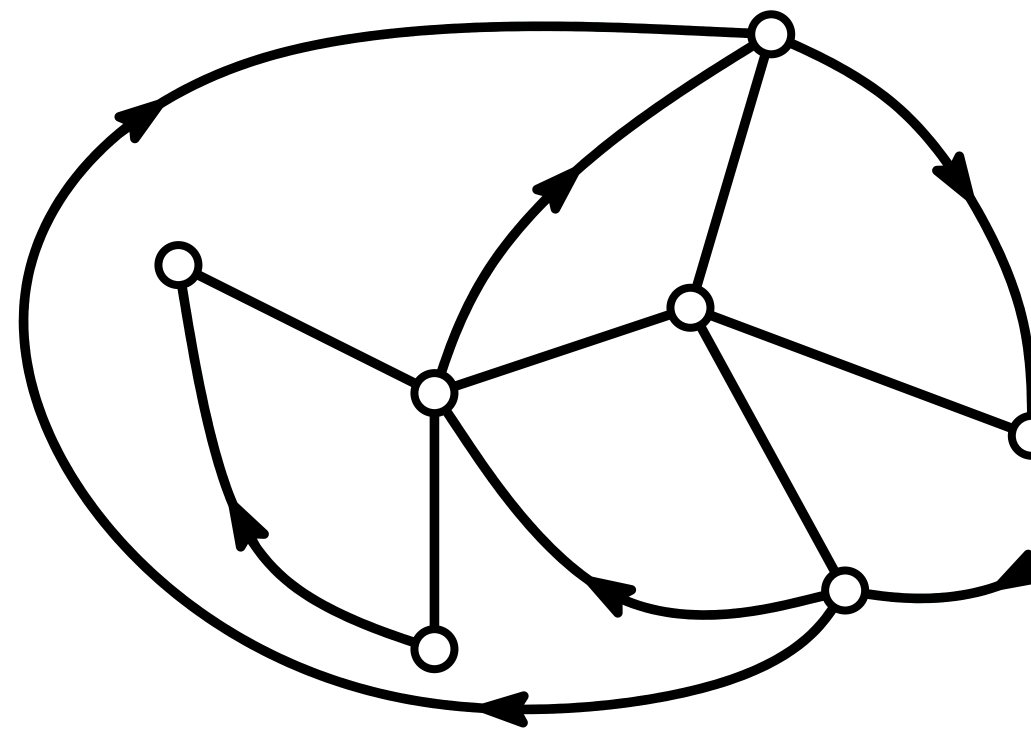
$$\#\text{ closing stems} = \#\text{ opening stems}$$



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

What is a blossoming tree ?

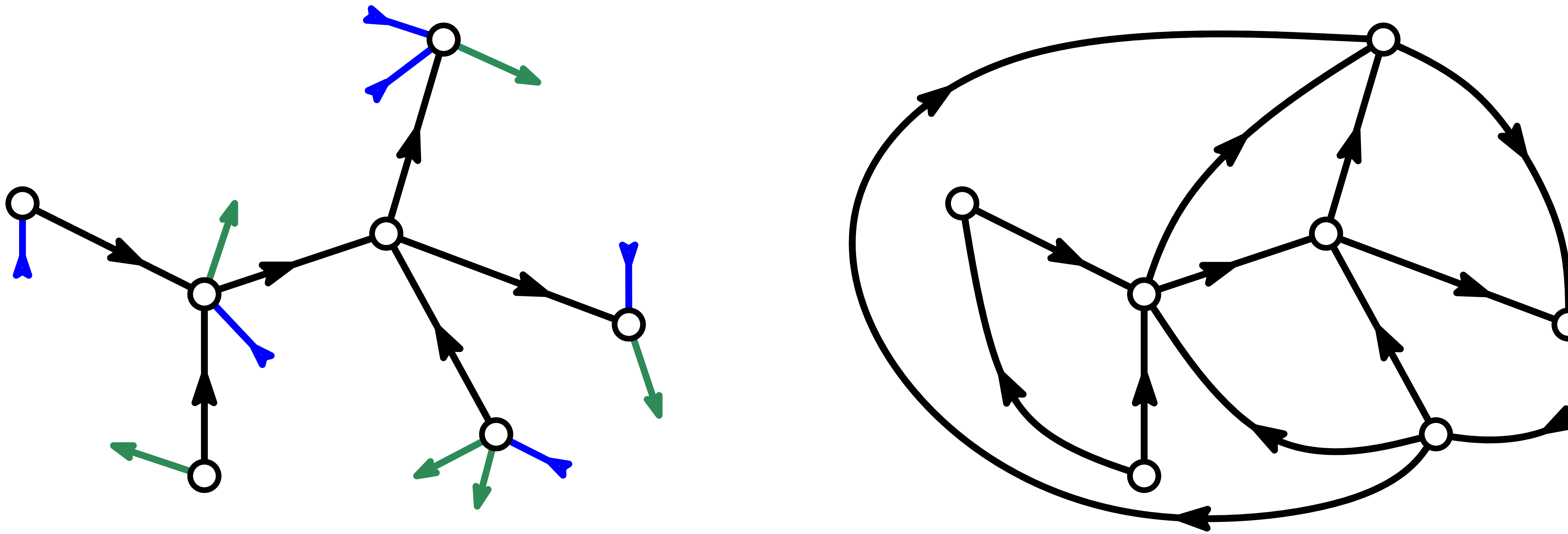




A plane map can be canonically associated to any blossoming tree by making all closures clockwise.

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 + closure edges oriented naturally
⇒ **orientation of the map without ccw cycles**.

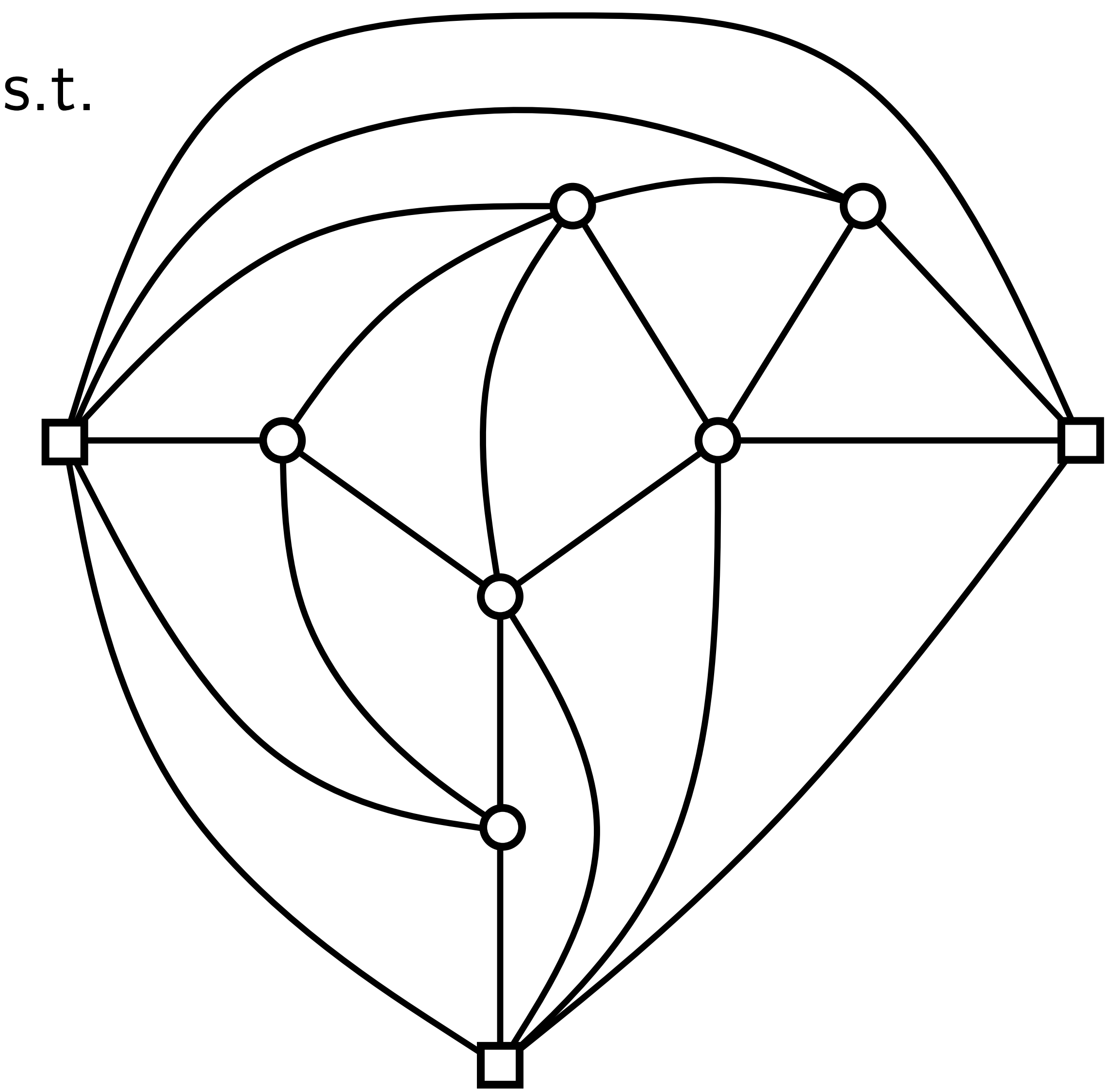
Orientations

First: orientation for **simple triangulations**

3-orientation $=$ orientation of the edges s.t.

out(□) $= 1$

out(○) $= 3$



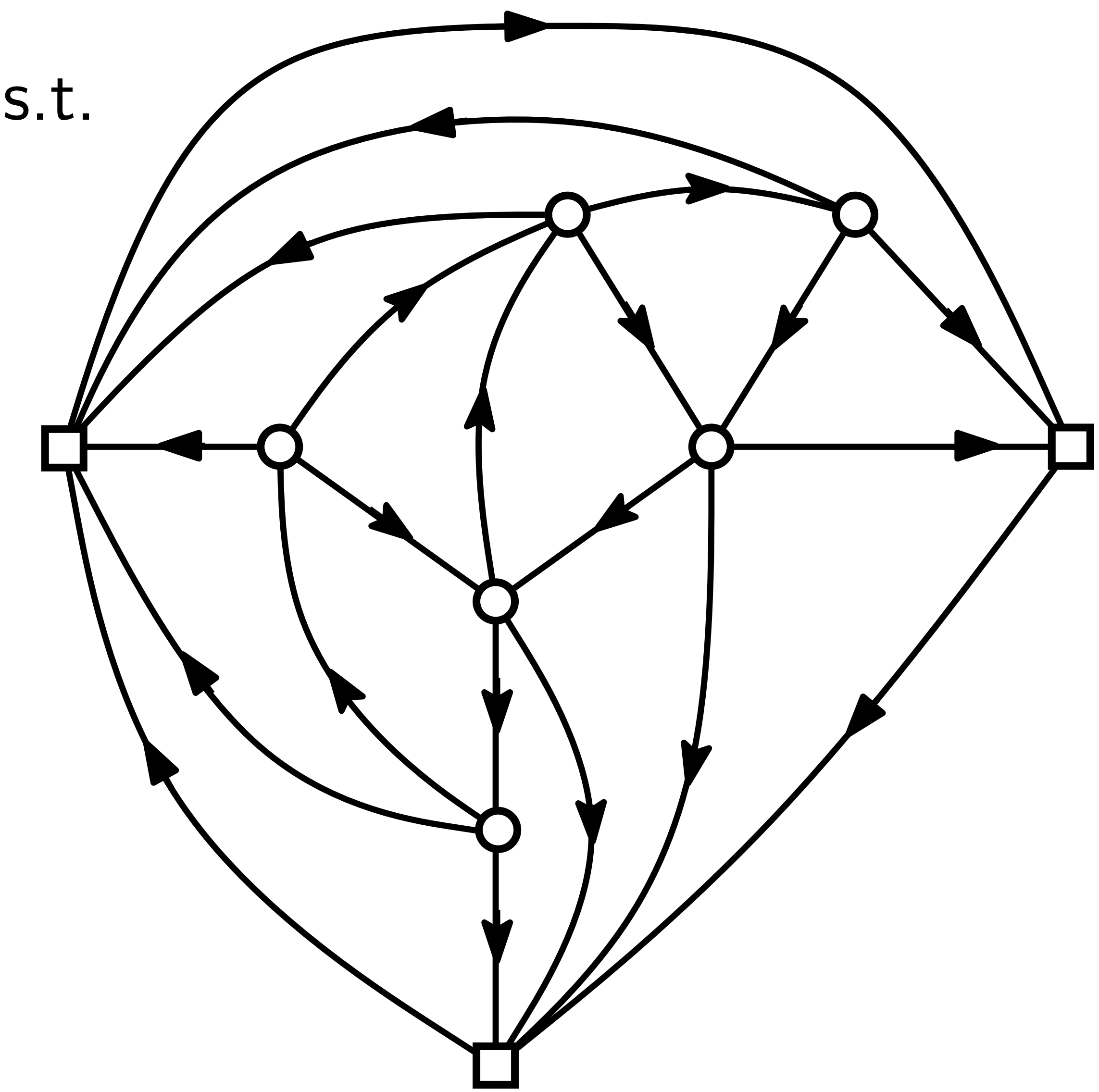
Orientations

First: orientation for **simple triangulations**

3-orientation $=$ orientation of the edges s.t.

out($\square$) $= 1$

out($\bigcirc$) $= 3$



Orientations

First: orientation for **simple triangulations**

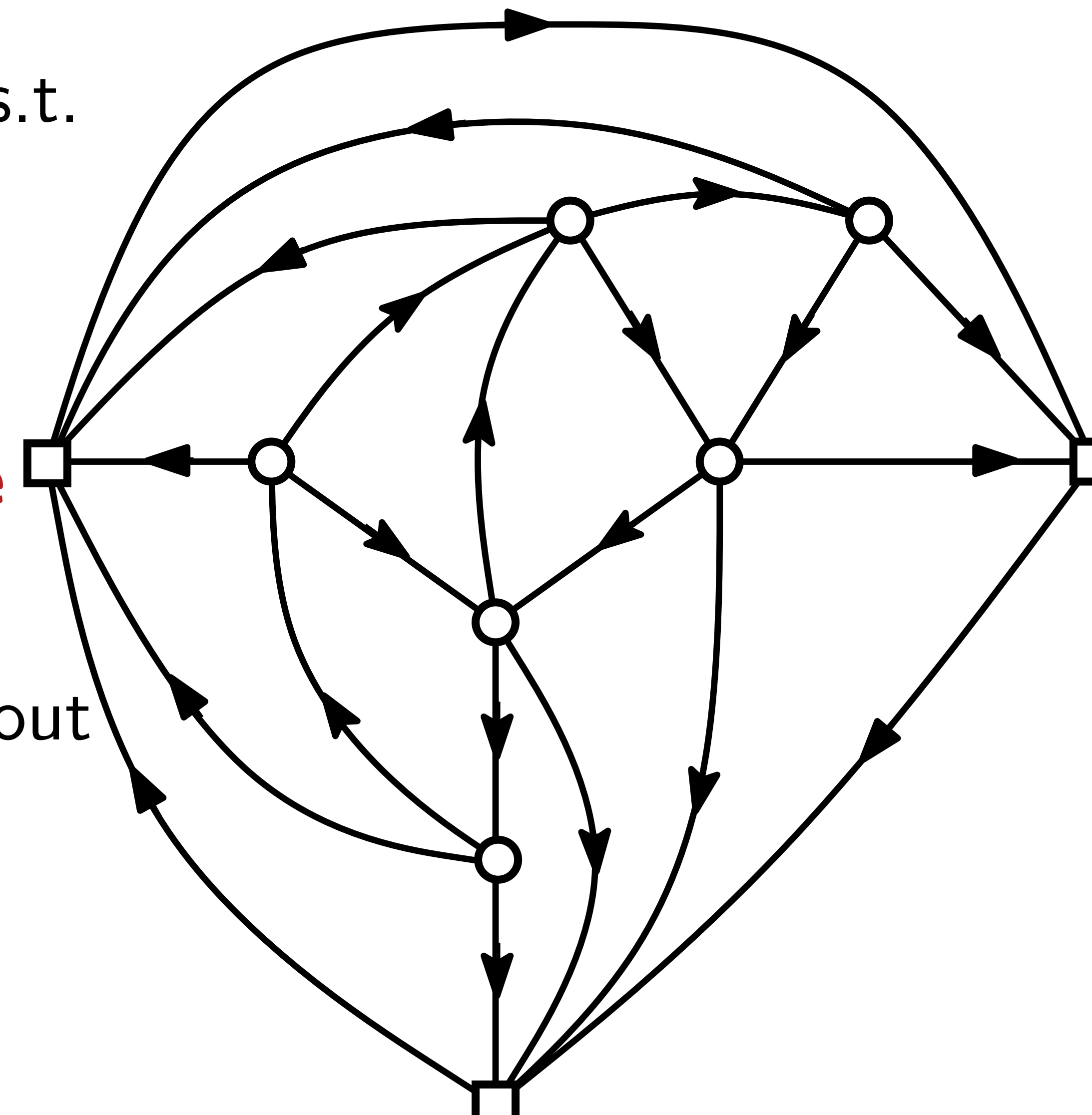
3-orientation $=$ orientation of the edges s.t.

$\text{out}(\square) = 1$

$\text{out}(\circ) = 3$

These orientations characterize simple triangulations [Schnyder]

Moreover, there exists a **unique** one without counterclockwise cycles.



Orientations

First: orientation for **simple triangulations**

3-orientation = orientation of the edges s.t.

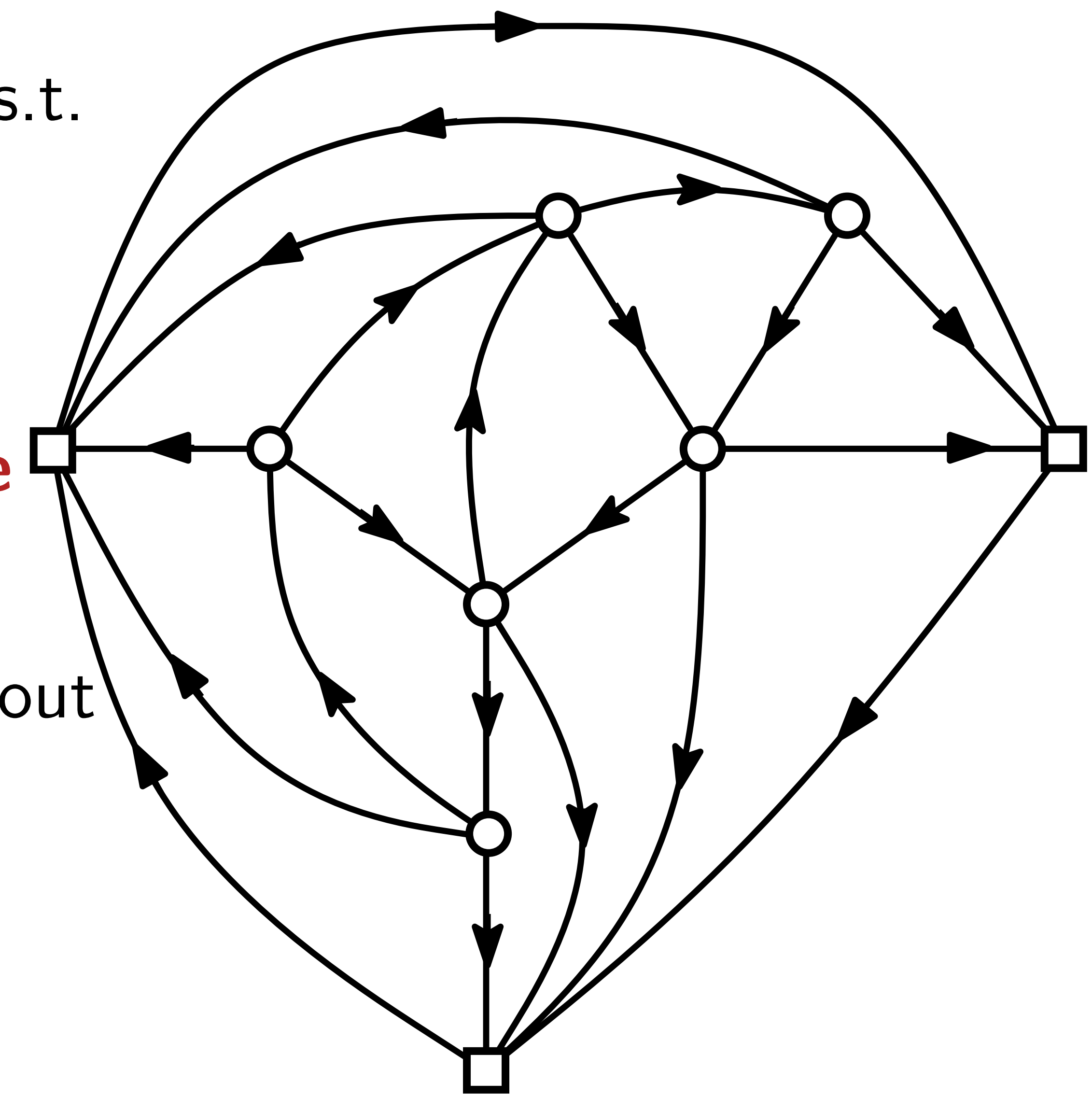
out(□) = 1

out(○) = 3

These orientations characterize simple triangulations [Schnyder]

Moreover, there exists a **unique** one without counterclockwise cycles.

What about **general simple maps ?**



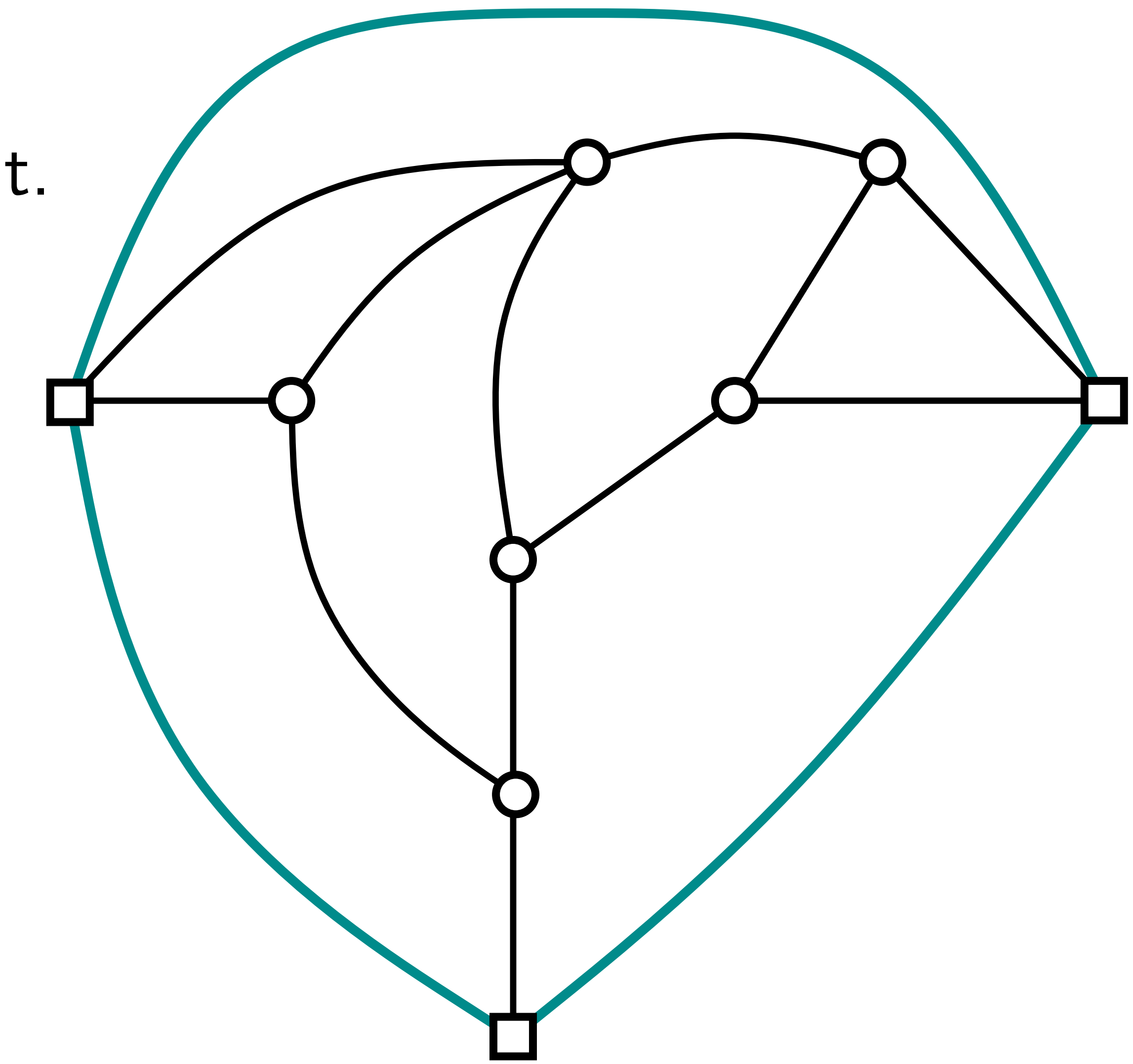
Orientations

Next: orientation for **simple outer-triangular maps**

3-orientation with buds $=$
orientation of the edges and of **buds** s.t.

out(□) $= 1$

out(○) $= 3$



Orientations

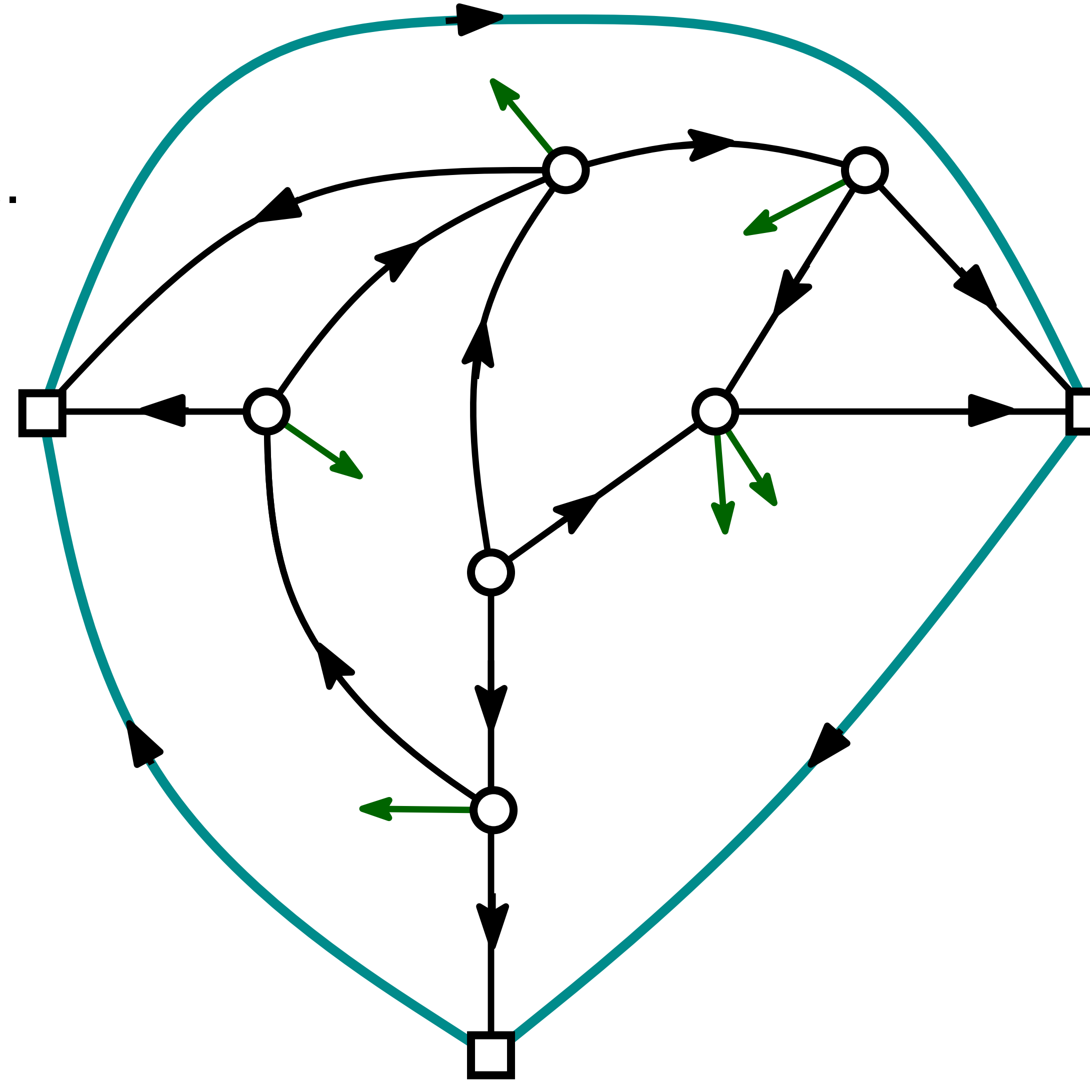
Next: orientation for **simple outer-triangular maps**

3-orientation with buds $=$
orientation of the edges and of **buds** s.t.

out(□) $= 1$

out(○) $= 3$

and each face of degree d is incident to $d-3$ buds.



Orientations

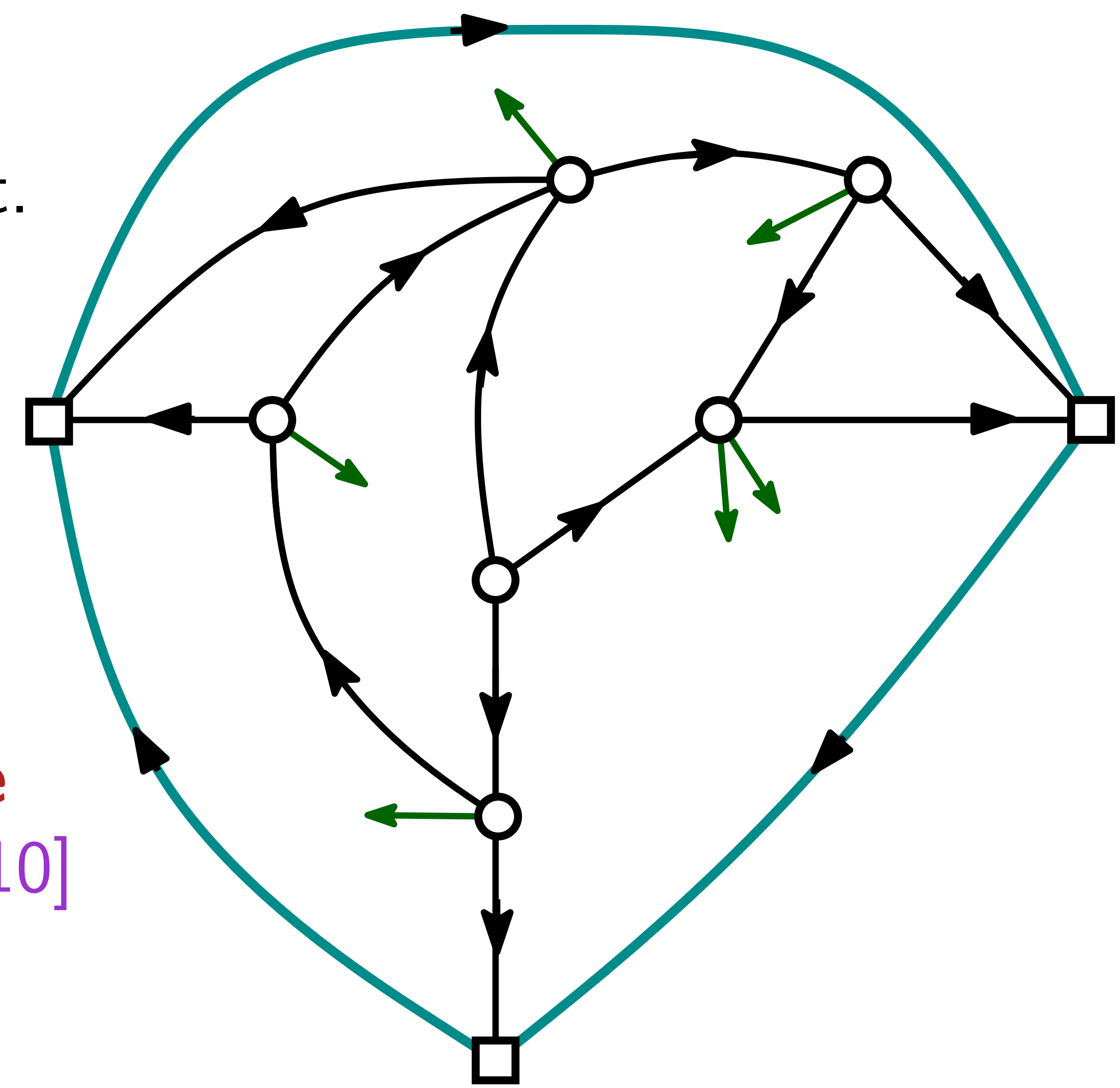
Next: orientation for **simple outer-triangular maps**

3-orientation with buds $=$
orientation of the edges and of **buds** s.t.

out(□) $= 1$

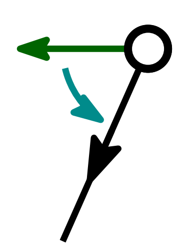
out(○) $= 3$

and each face of degree d is incident to $d-3$ buds.



These orientations characterize simple outer-triangular maps [Bernardi, Fusy '10]

Moreover, there exists a **unique** one without counterclockwise cycles and local configuration:



Orientations

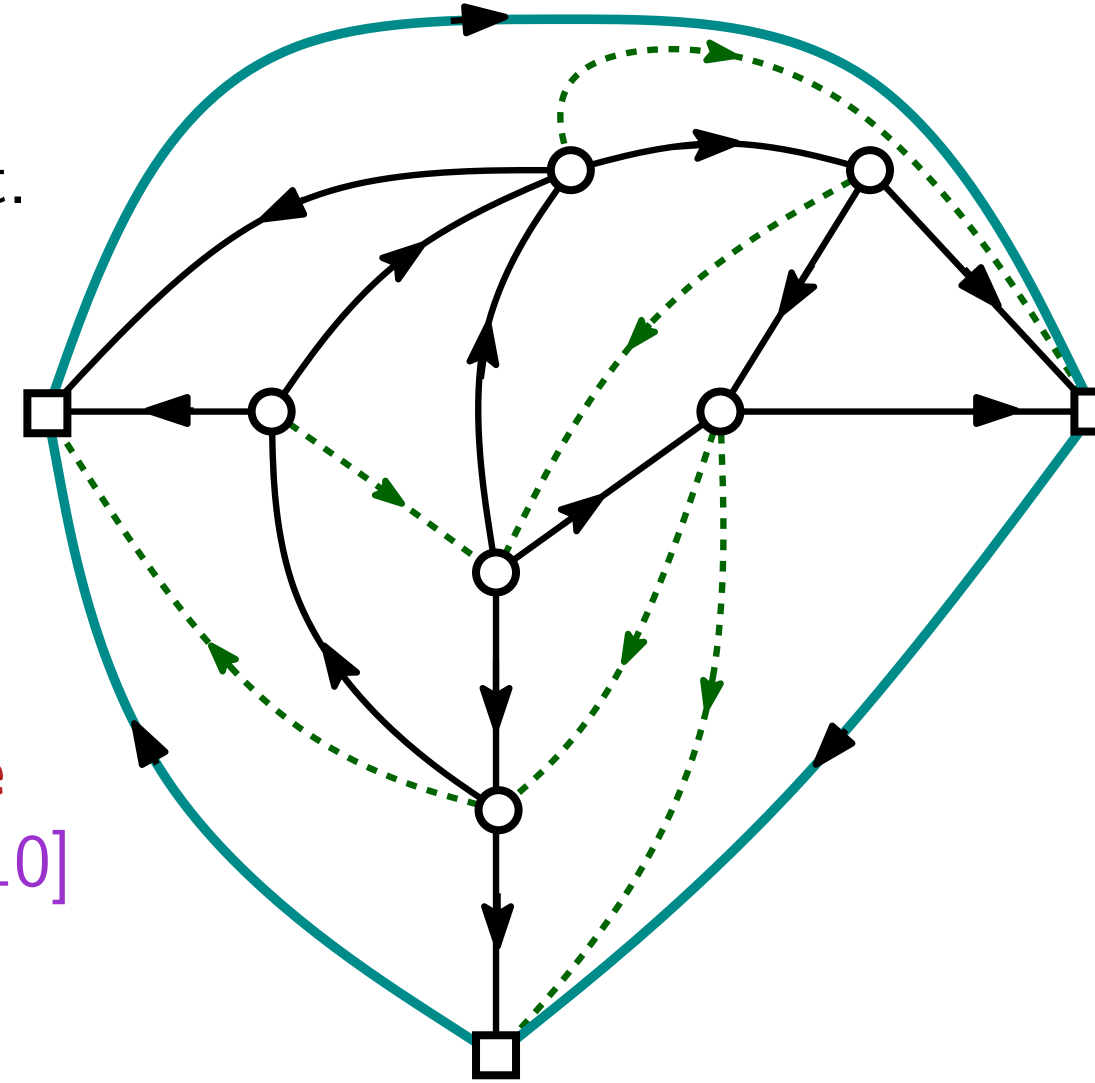
Next: orientation for **simple outer-triangular maps**

3-orientation with buds $=$
orientation of the edges and of **buds** s.t.

out(□) $= 1$

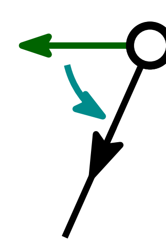
out(○) $= 3$

and each face of degree d is incident to $d-3$ buds.



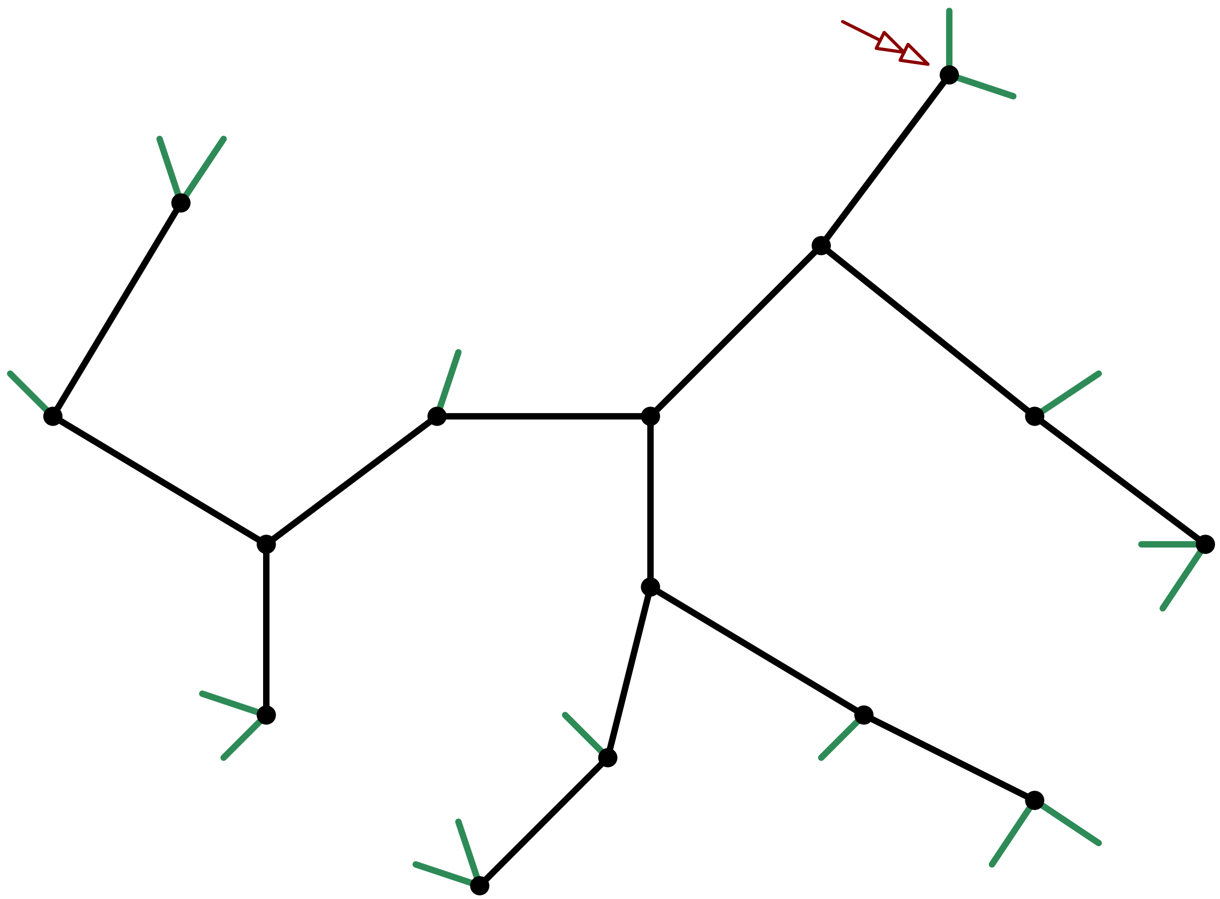
These orientations characterize simple outer-triangular maps [Bernardi, Fusy '10]

Moreover, there exists a **unique** one without counterclockwise cycles and local configuration:



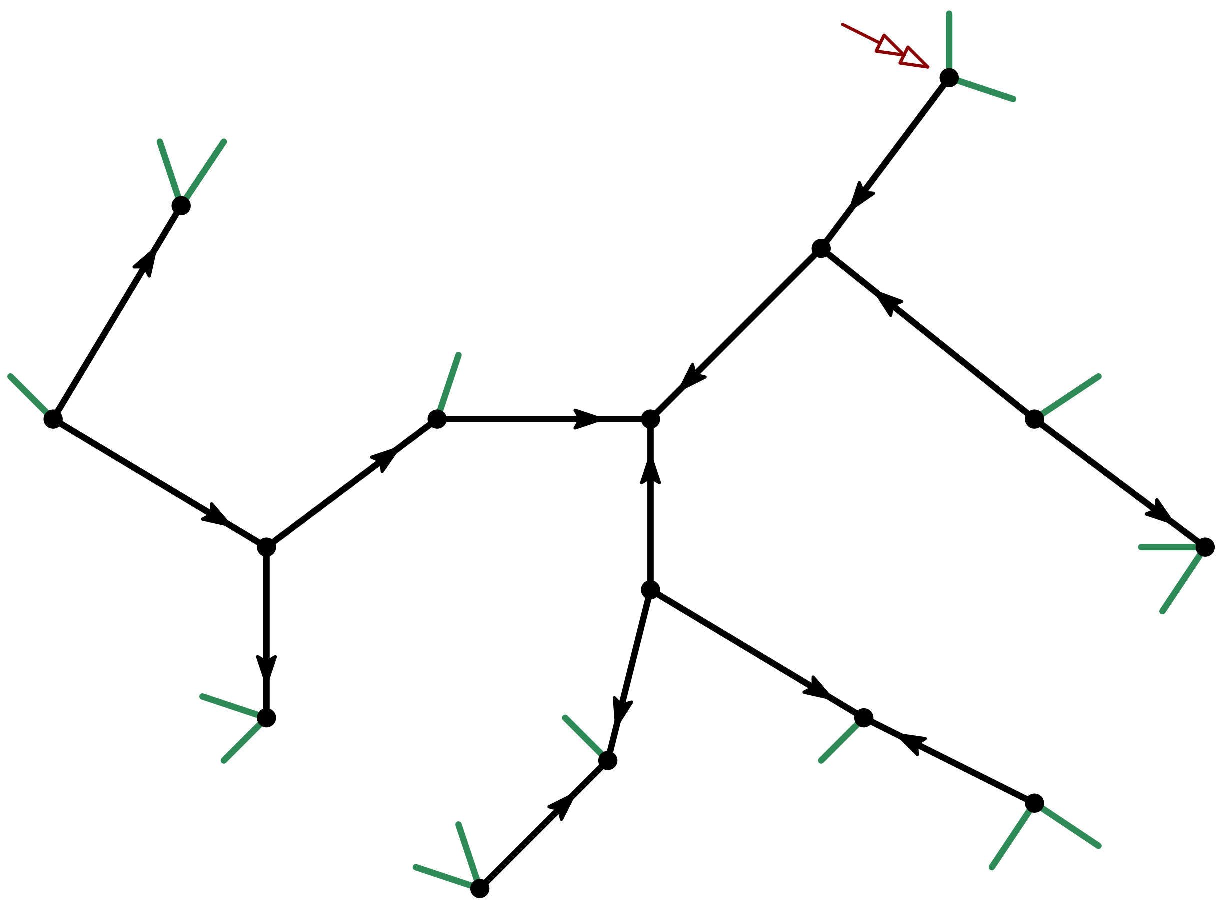
⇒ Give a canonical triangulation of a simple map

Oriented binary trees



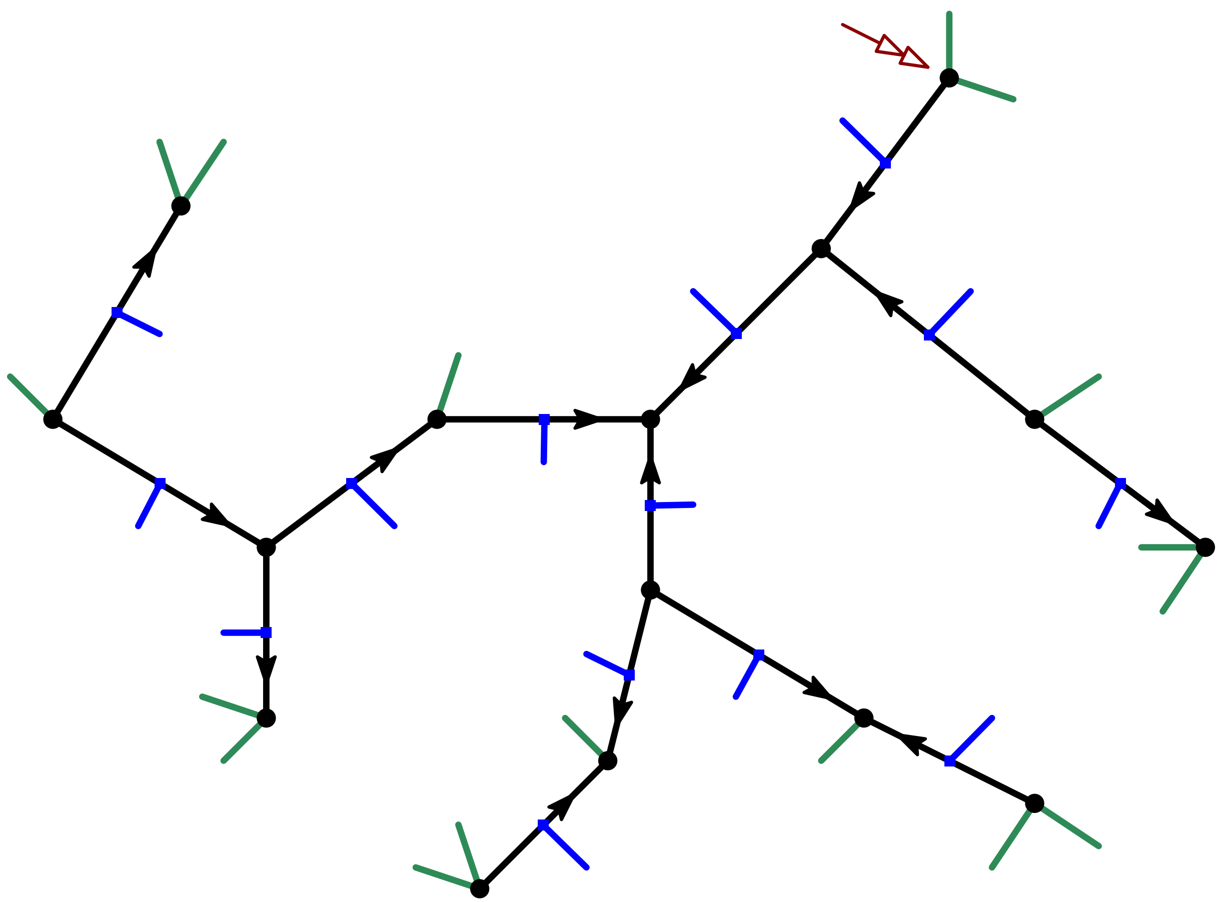
- Start from a rooted binary tree (each inner vertex has 3 neighbours)

Oriented binary trees

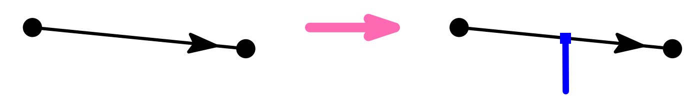


- Start from a rooted binary tree (each inner vertex has 3 neighbours)
- Orient each inner edge arbitrarily

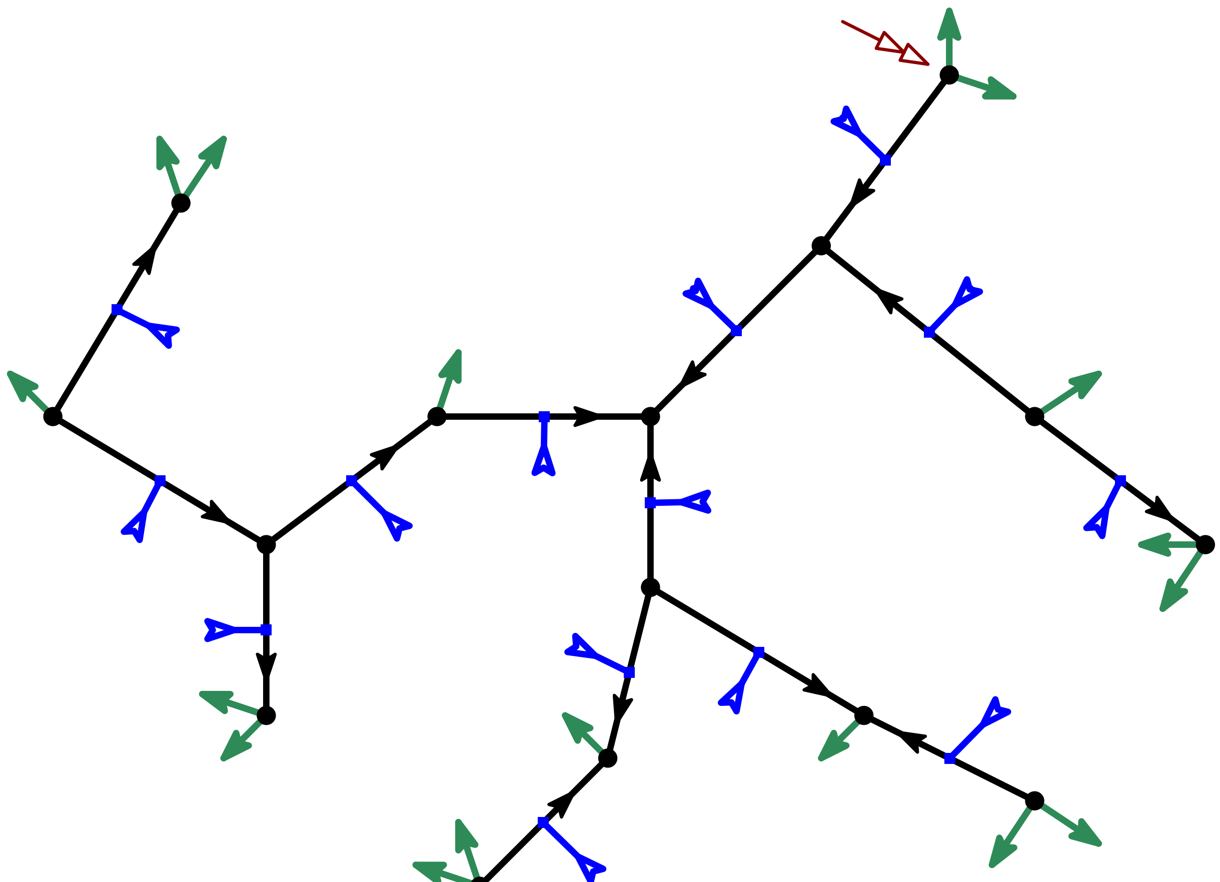
Oriented binary trees



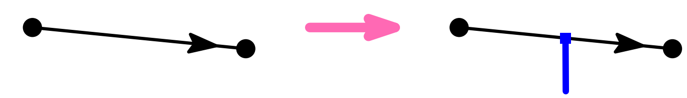
- Start from a rooted binary tree (each inner vertex has 3 neighbours)
- Orient each inner edge arbitrarily
- Apply the following local rule :



Oriented binary trees

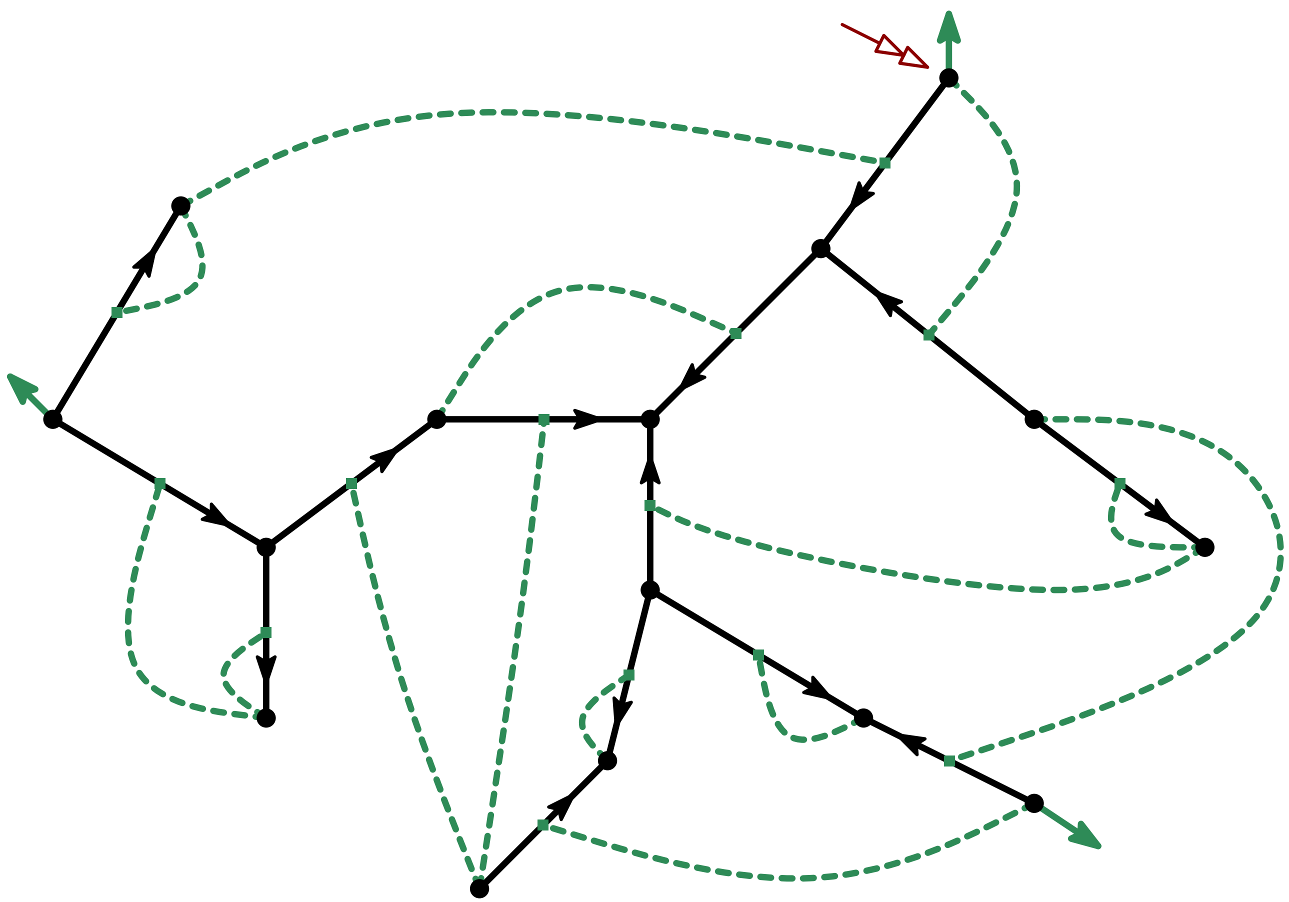


- Start from a rooted binary tree (each inner vertex has 3 neighbours)
- Orient each inner edge arbitrarily
- Apply the following local rule :

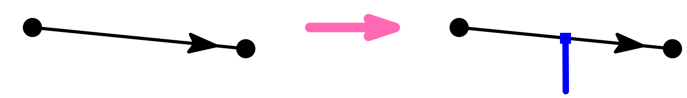


- See — as opening stems and — as closing stems.

Oriented binary trees

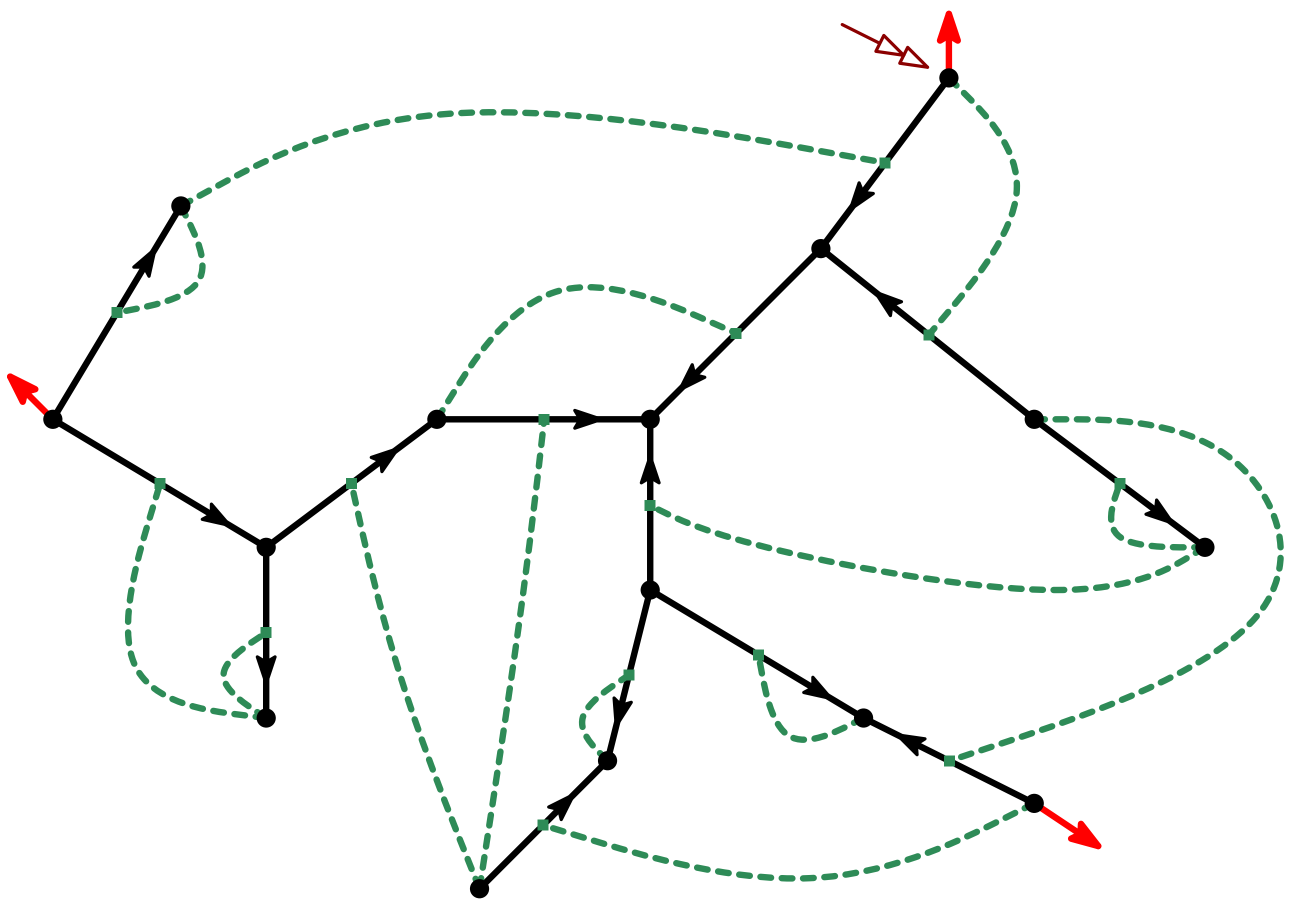


- Start from a rooted binary tree (each inner vertex has 3 neighbours)
- Orient each inner edge arbitrarily
- Apply the following local rule :

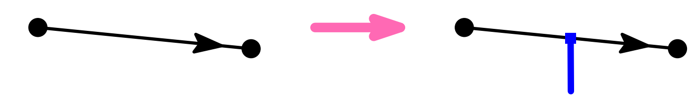


- See as opening stems and as closing stems.
- Make the closure

Oriented binary trees

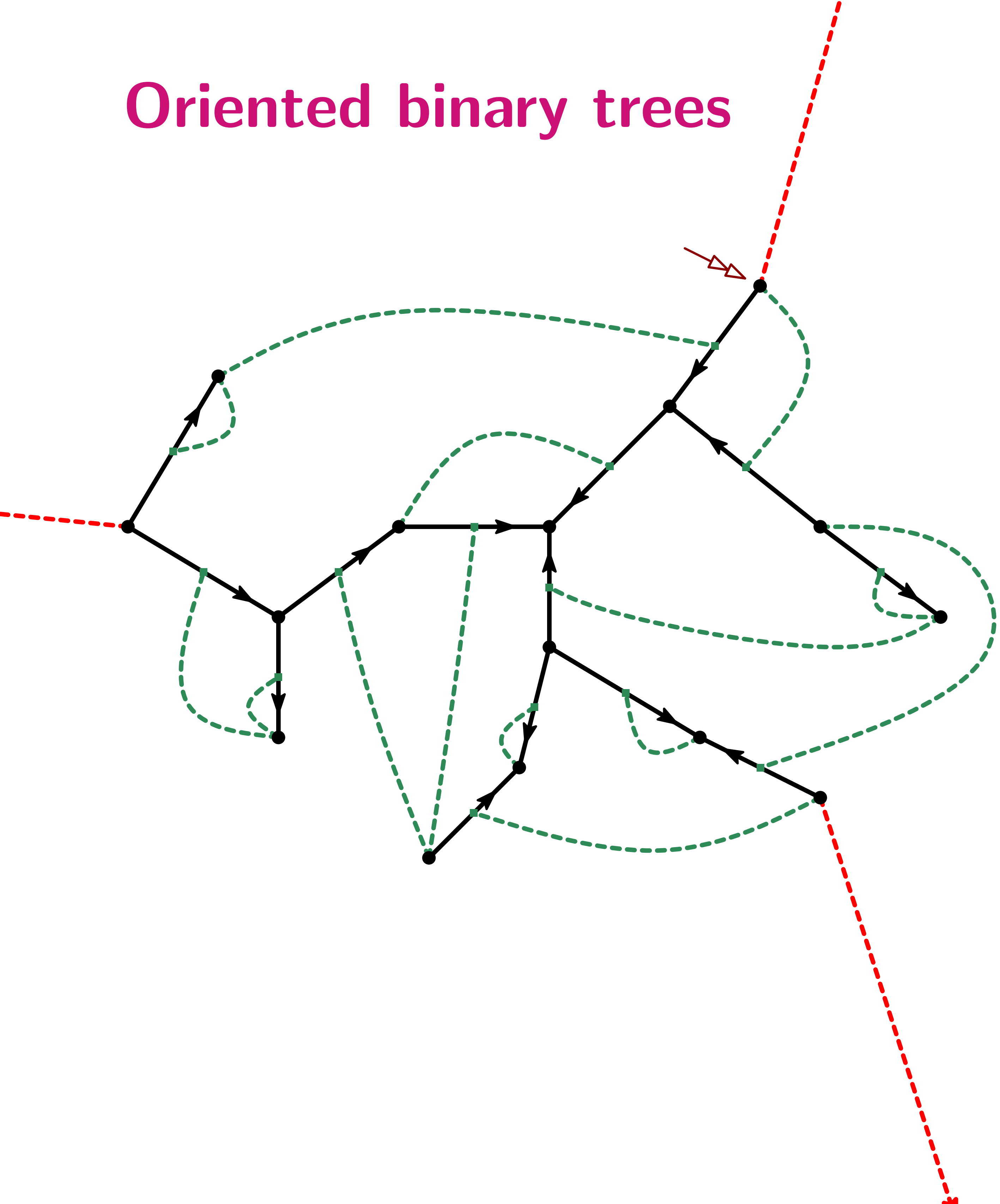


- Start from a rooted binary tree (each inner vertex has 3 neighbours)
- Orient each inner edge arbitrarily
- Apply the following local rule :

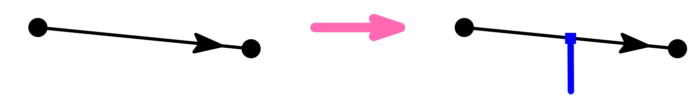


- See as opening stems and as closing stems.
- Make the closure
- 3 opening stems are left unmatched,
- one of them is incident to the root corner: the tree is **balanced**

Oriented binary trees

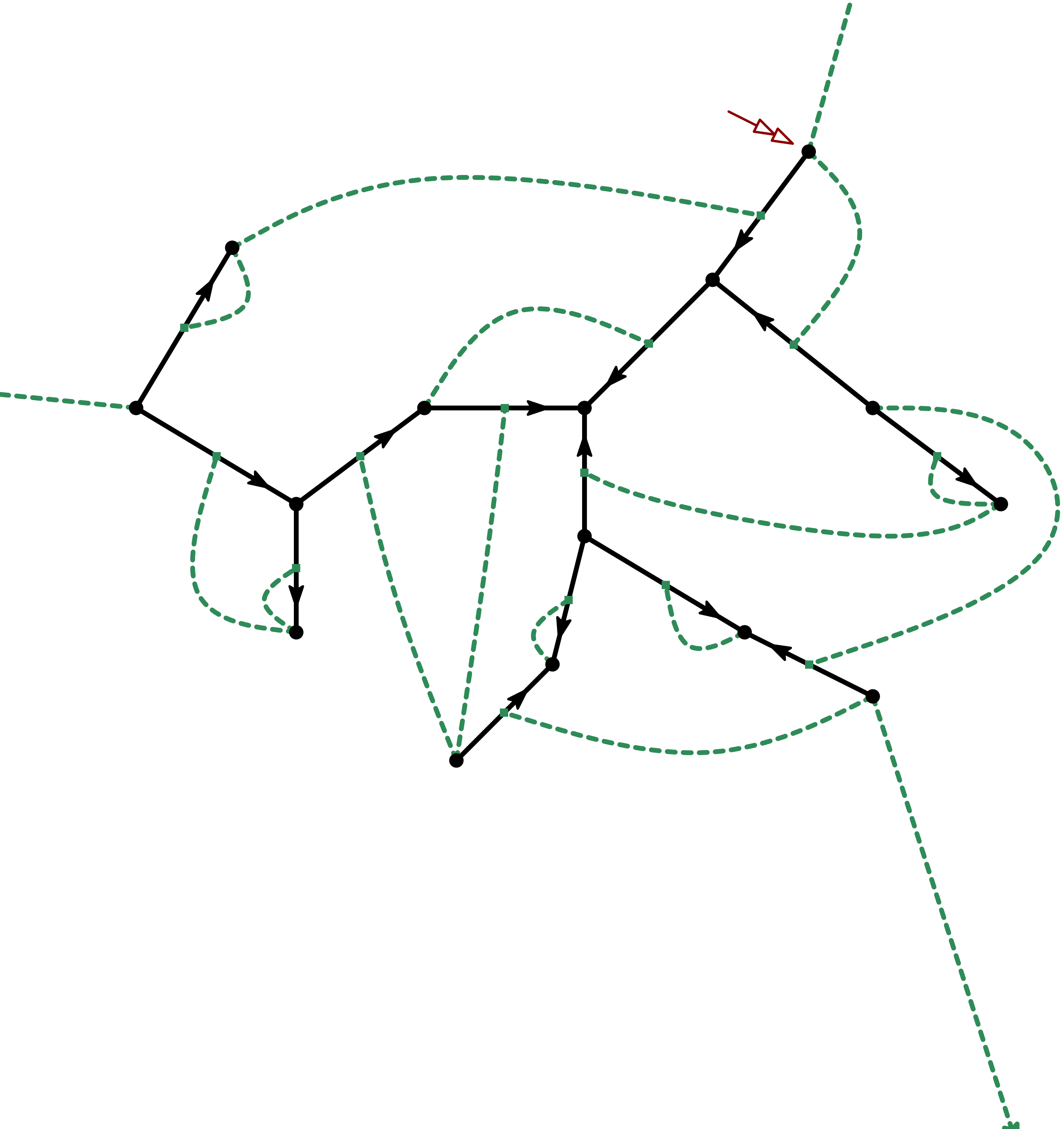


- Start from a rooted binary tree (each inner vertex has 3 neighbours)
- Orient each inner edge arbitrarily
- Apply the following local rule :

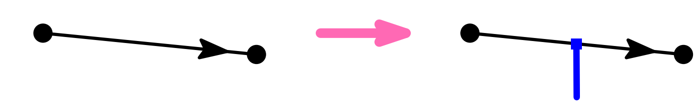


- See as opening stems and as closing stems.
- Make the closure
- 3 opening stems are left unmatched, we close them at ∞
- one of them is incident to the root corner: the tree is **balanced**

Oriented binary trees

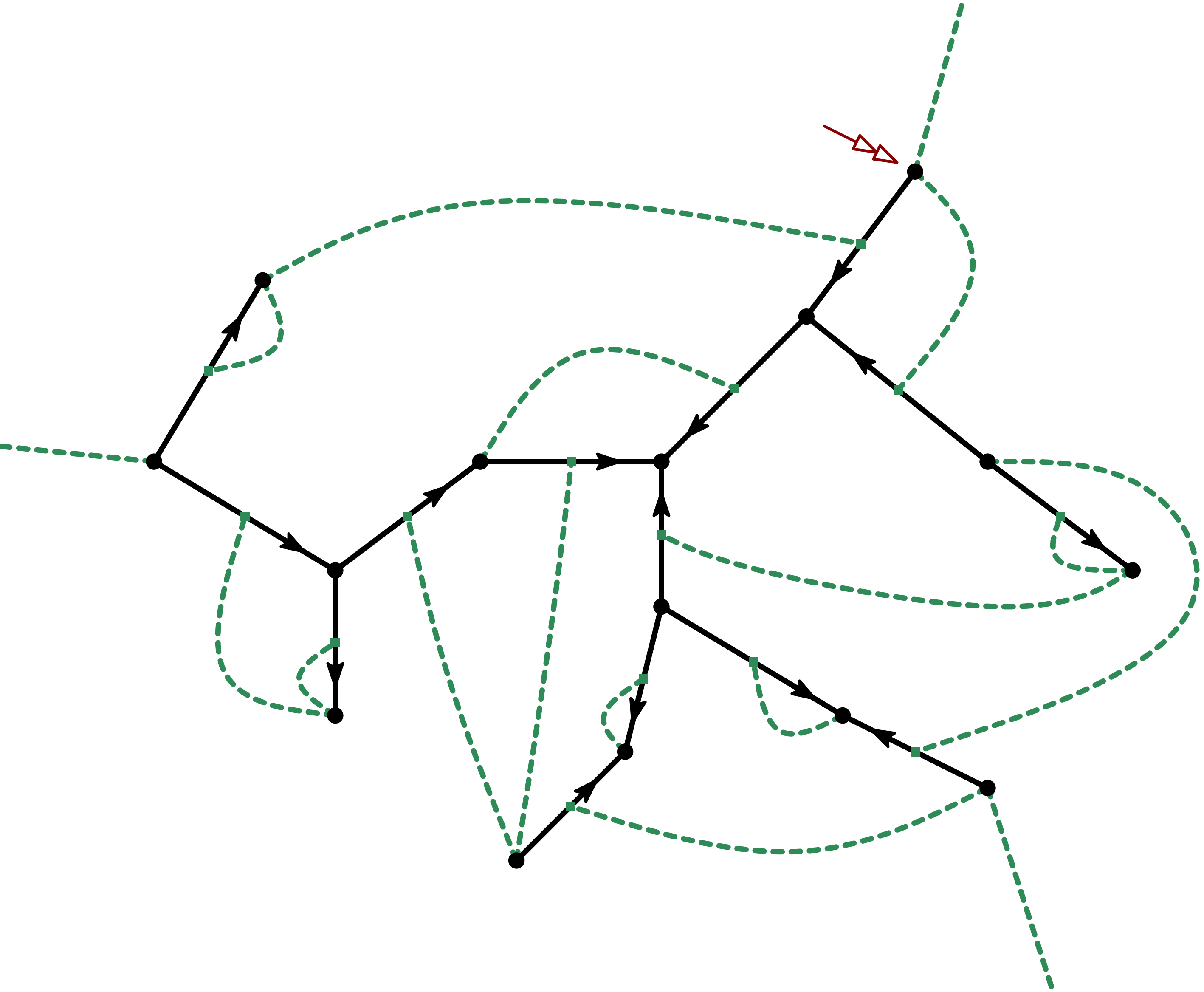


- Start from a rooted binary tree (each inner vertex has 3 neighbours)
- Orient each inner edge arbitrarily
- Apply the following local rule :

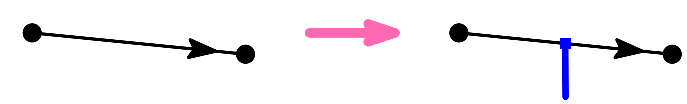


- See —— as opening stems and —— as closing stems.
- Make the closure
- 3 opening stems are left unmatched, we close them at ∞
- one of them is incident to the root corner: the tree is **balanced**
  - We obtain a bipartite cubic map

Oriented binary trees



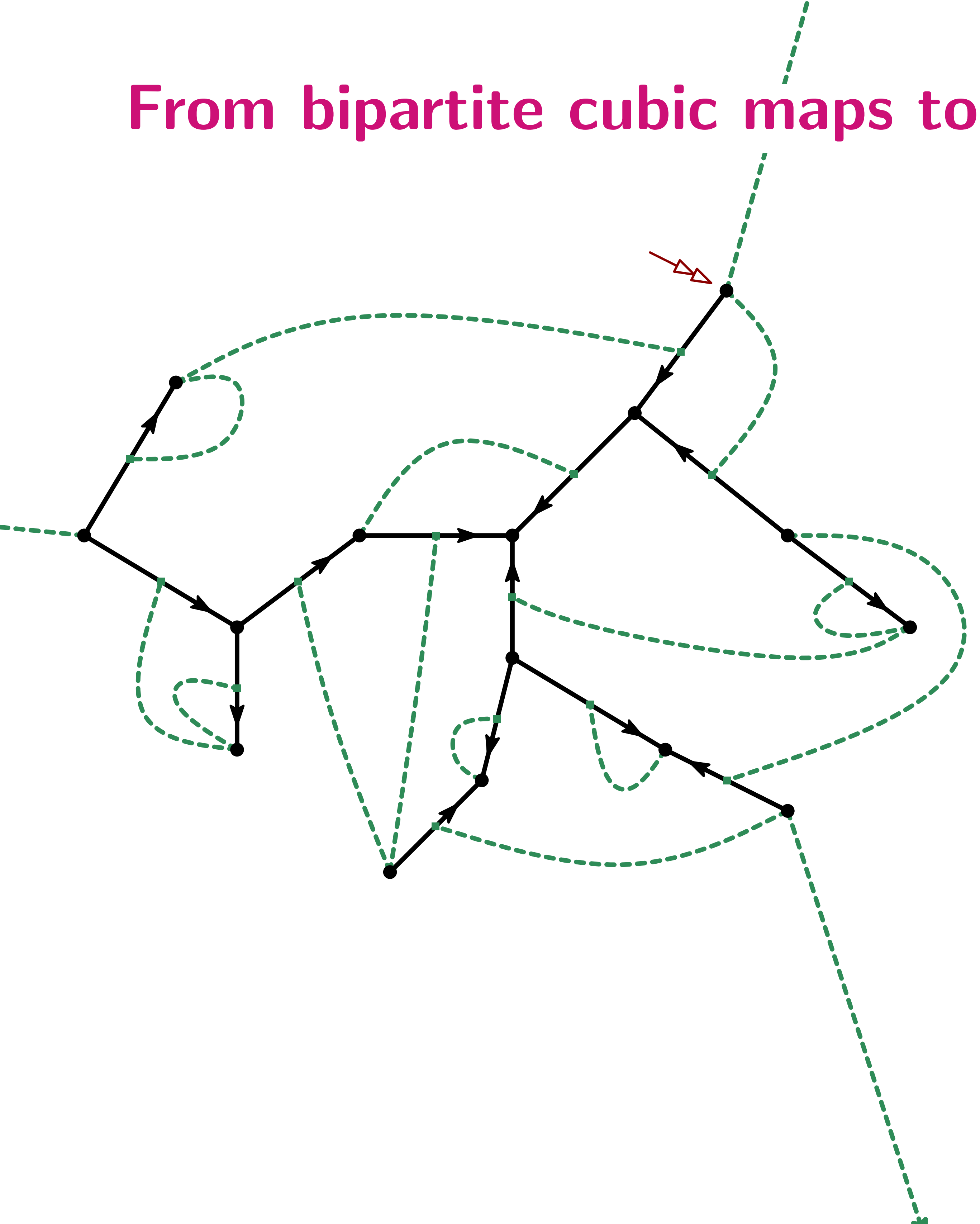
- Start from a rooted binary tree (each inner vertex has 3 neighbours)
- Orient each inner edge arbitrarily
- Apply the following local rule :



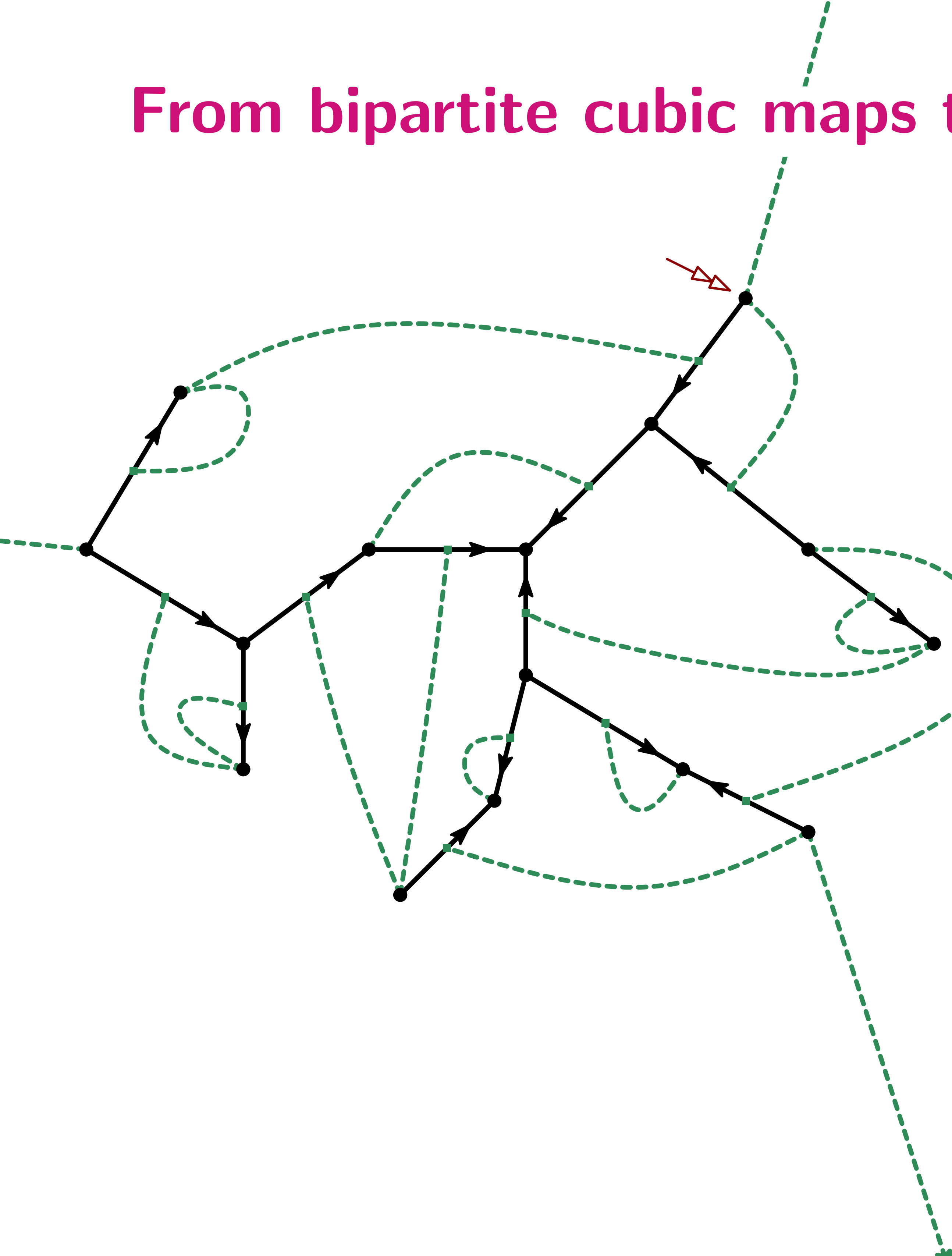
- See —— as opening stems and —— as closing stems.
- Make the closure
- 3 opening stems are left unmatched, we close them at ∞
- one of them is incident to the root corner: the tree is **balanced**

Theorem : [Bousquet-Mélou, Schaeffer '00] This is a bijection between **balanced oriented binary trees** and **bipartite cubic maps**

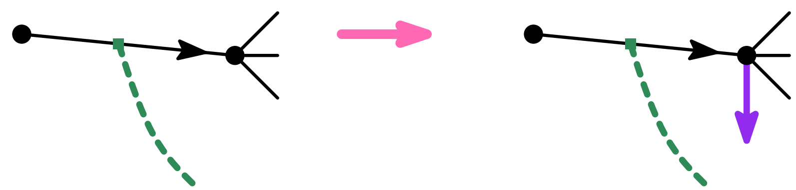
From bipartite cubic maps to simple maps



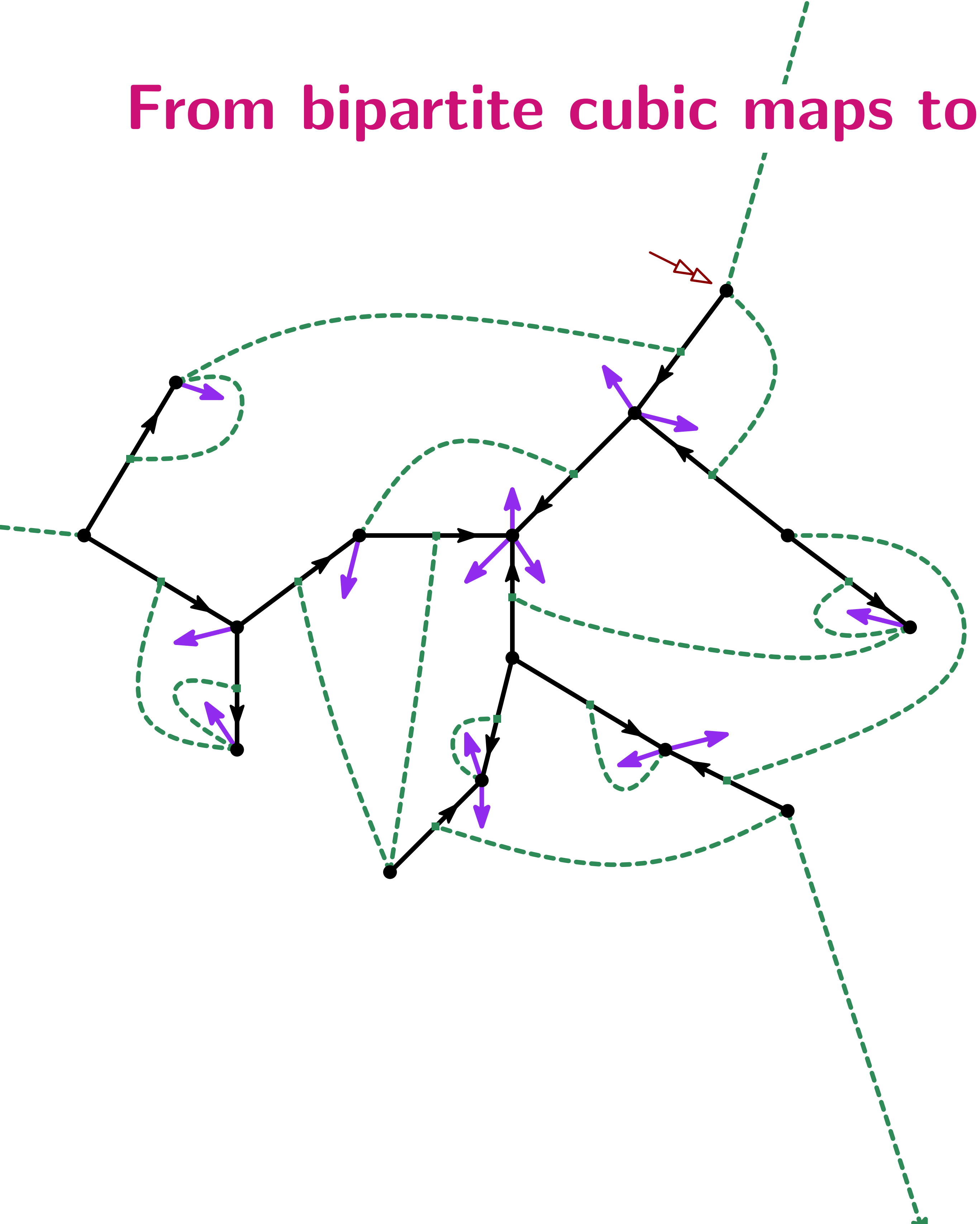
From bipartite cubic maps to simple maps



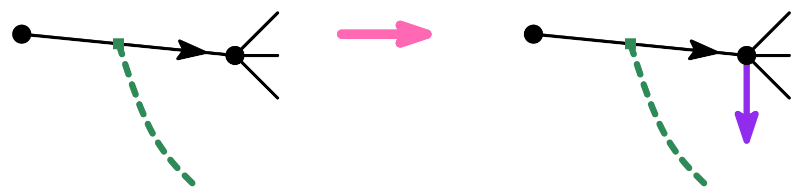
- Apply the following local rule :



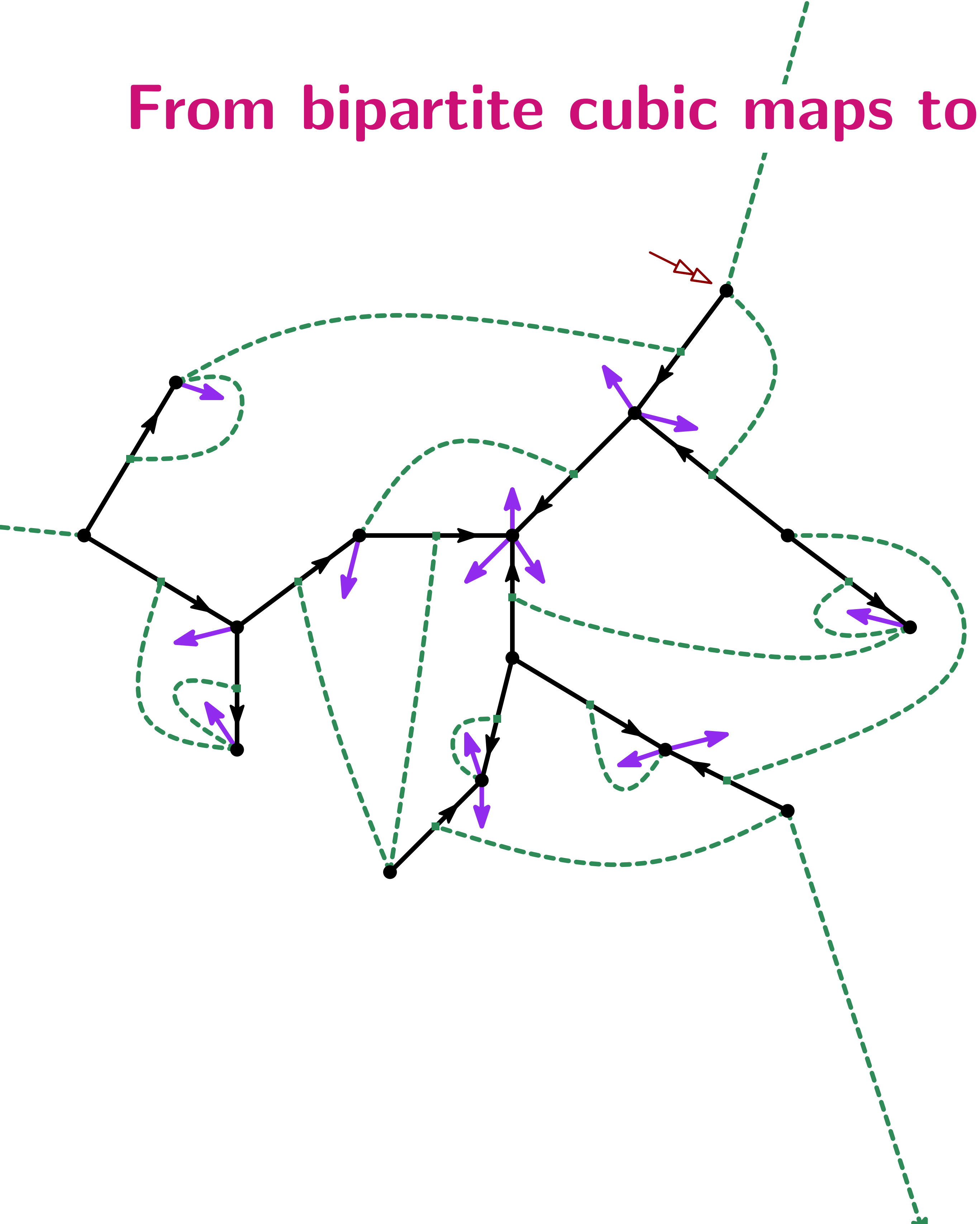
From bipartite cubic maps to simple maps



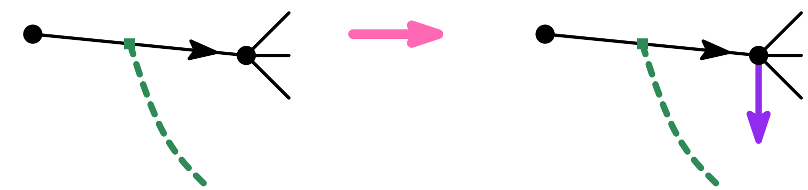
- Apply the following local rule :



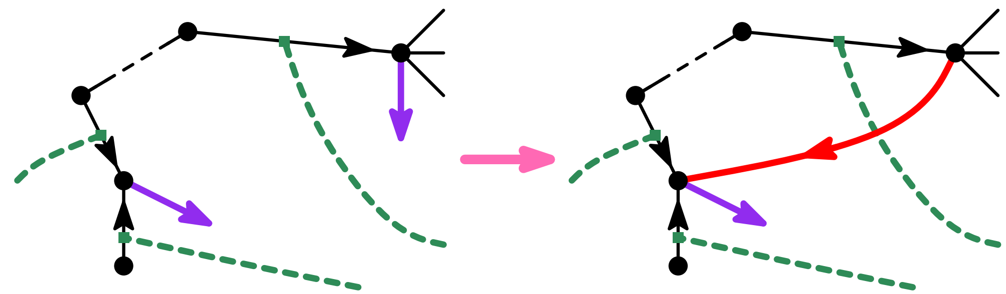
From bipartite cubic maps to simple maps



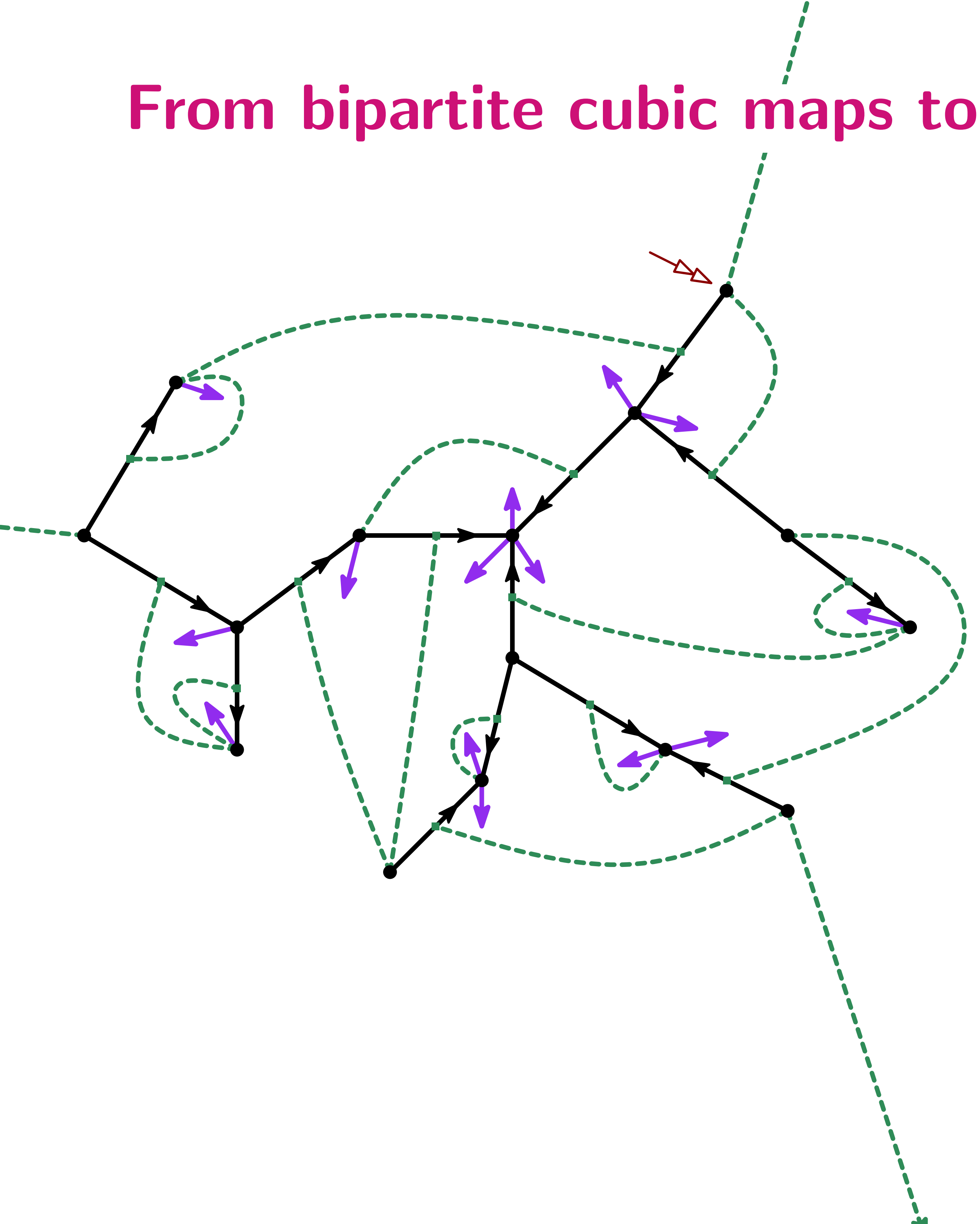
- Apply the following local rule :



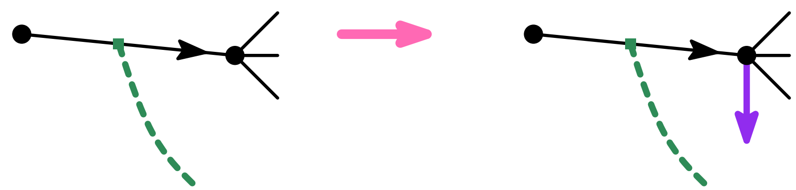
- Turning clockwise around the tree, do the following closures:



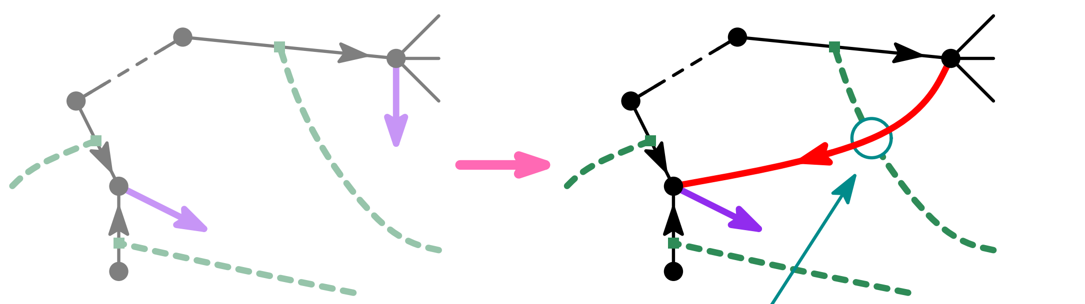
From bipartite cubic maps to simple maps



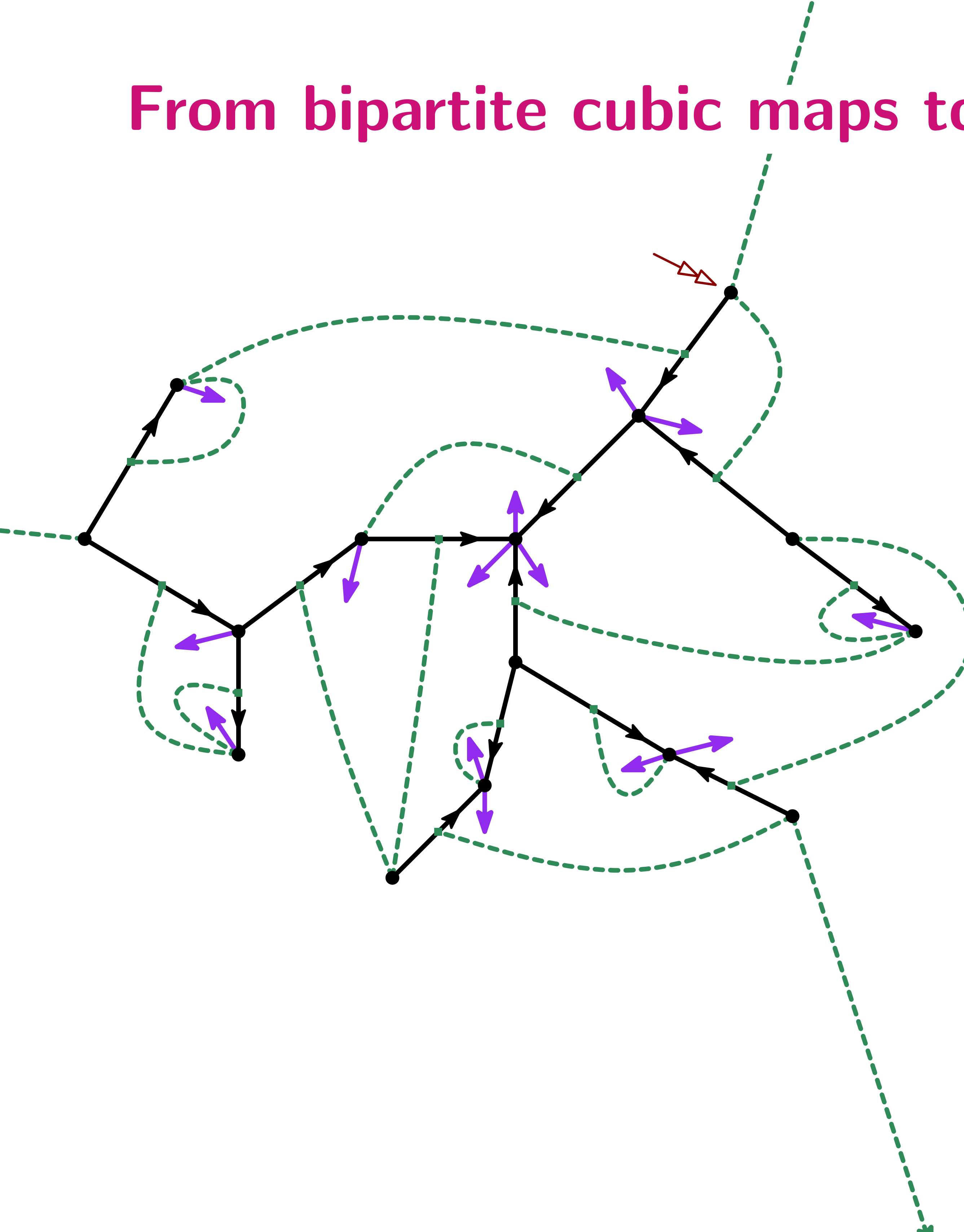
- Apply the following local rule :



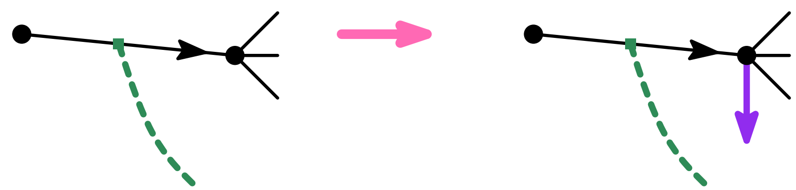
- Turning clockwise around the tree, do the following closures:



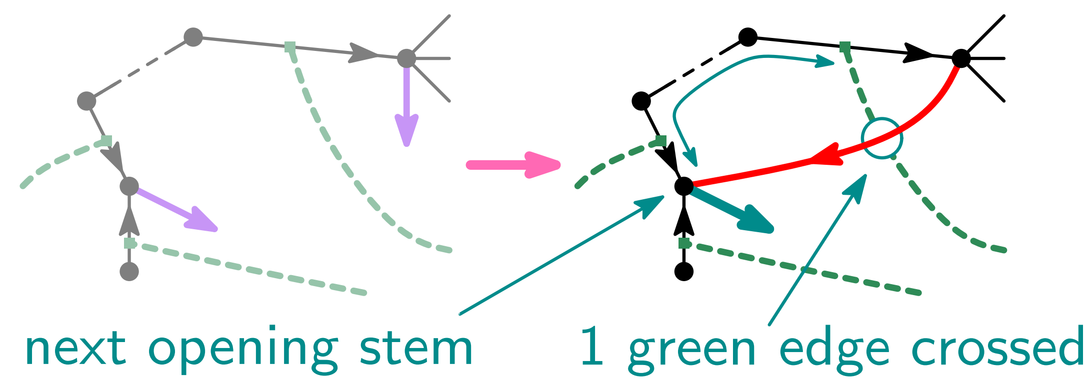
From bipartite cubic maps to simple maps



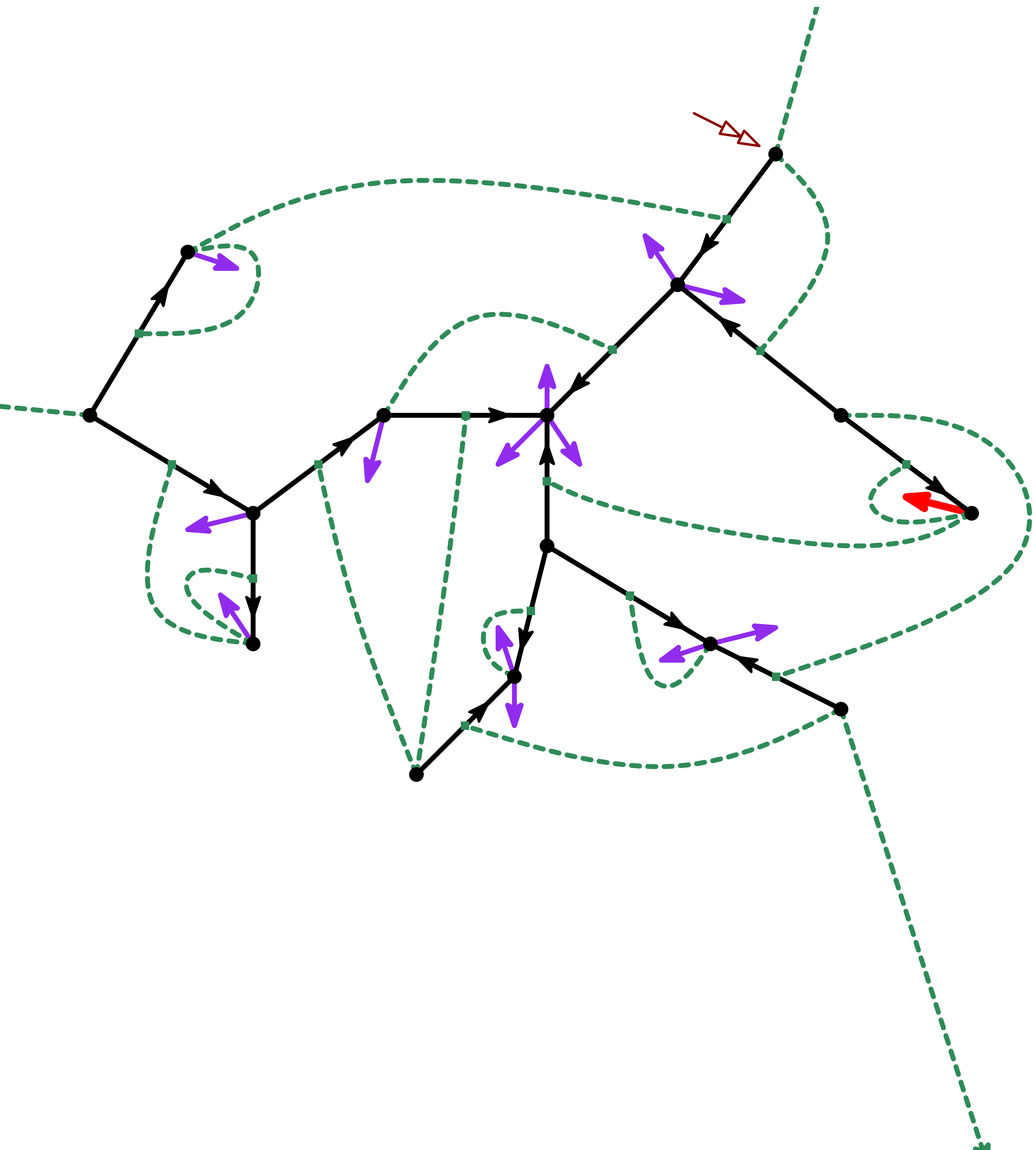
- Apply the following local rule :



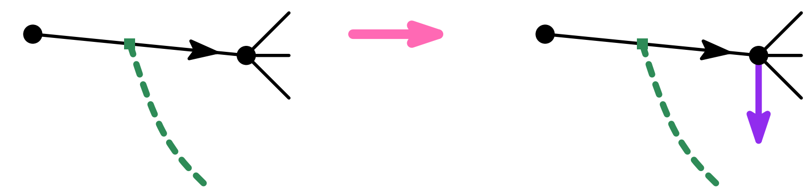
- Turning clockwise around the tree, do the following closures:



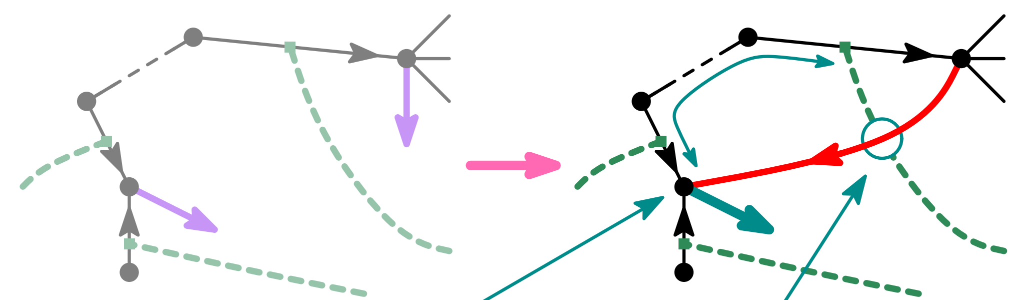
From bipartite cubic maps to simple maps



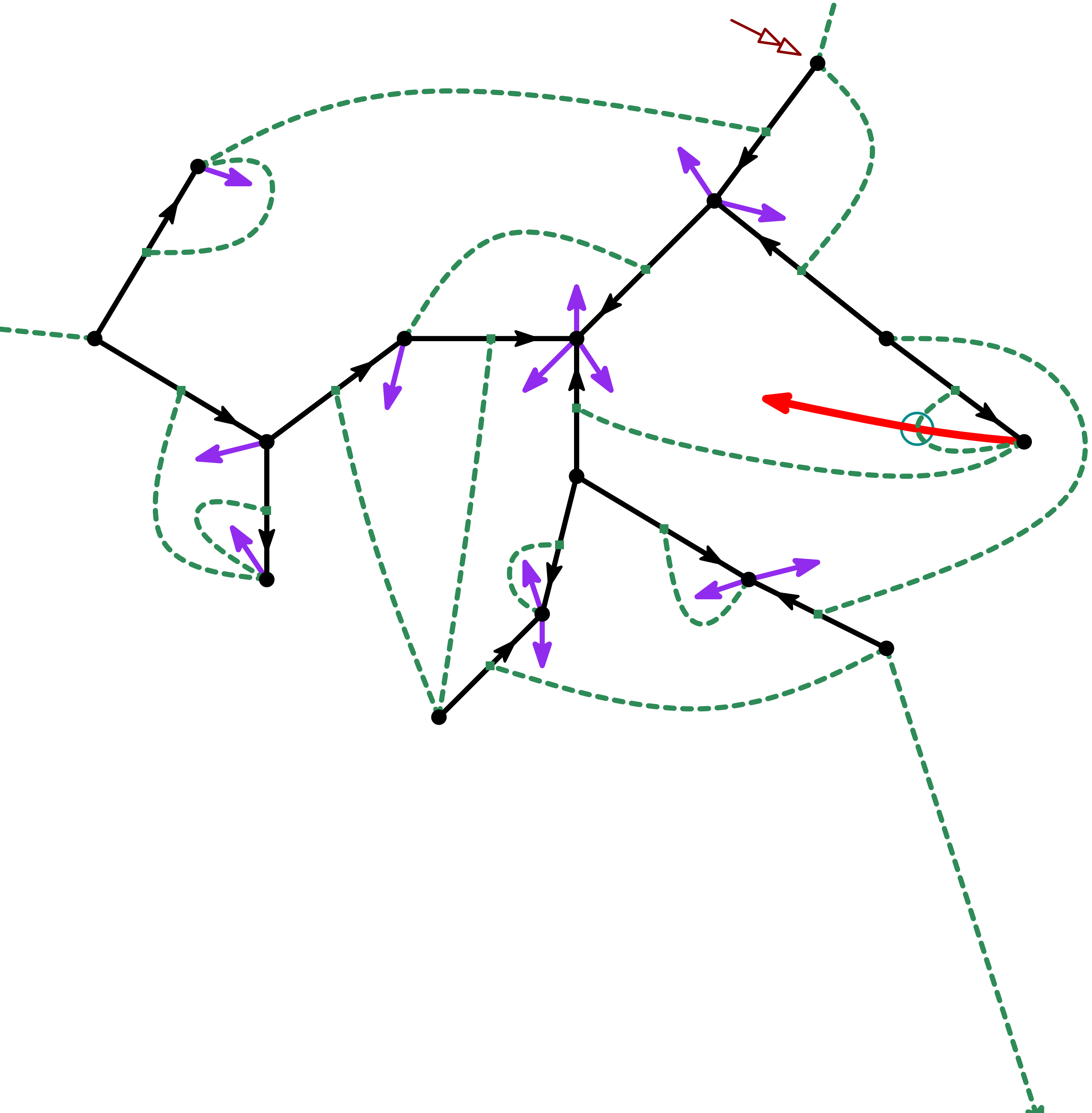
- Apply the following local rule :



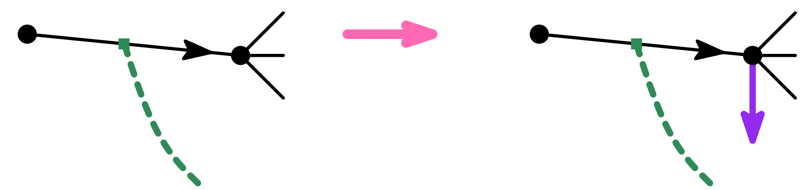
- Turning clockwise around the tree, do the following closures:



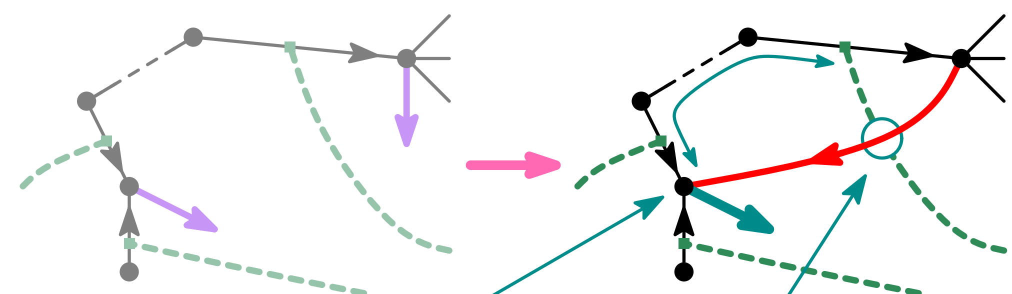
From bipartite cubic maps to simple maps



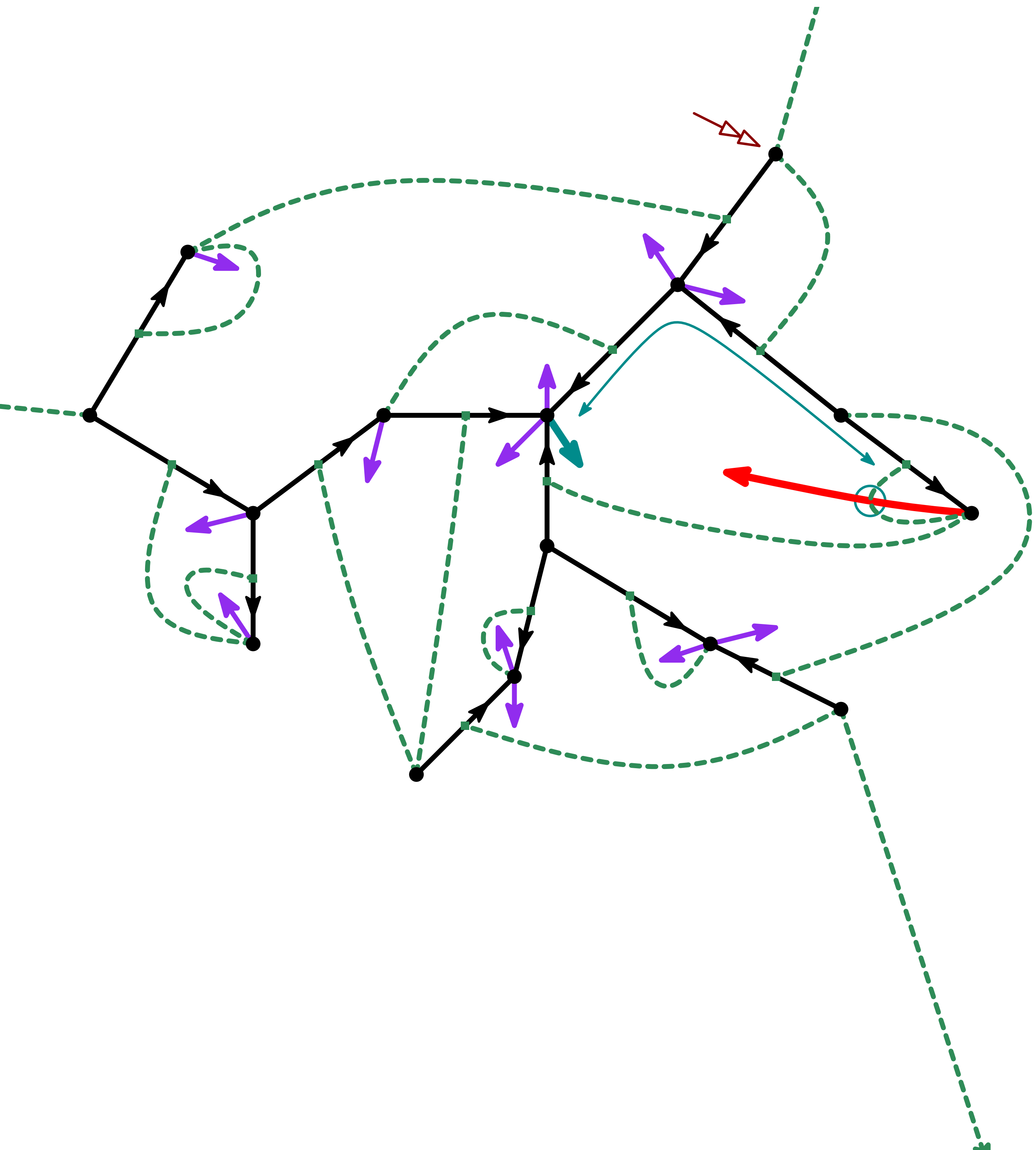
- Apply the following local rule :



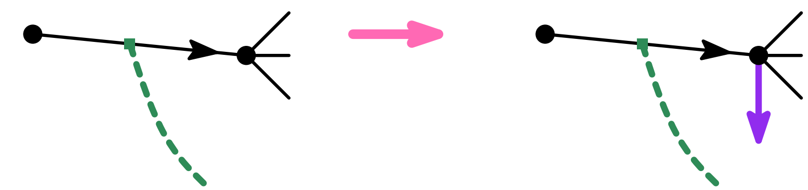
- Turning clockwise around the tree, do the following closures:



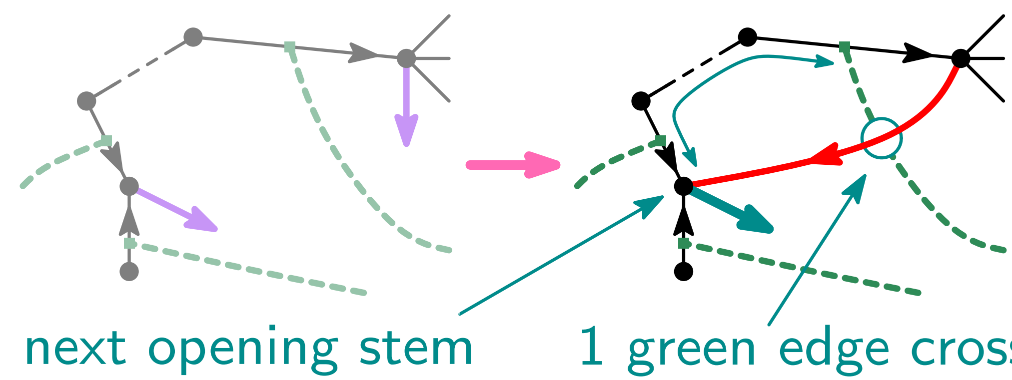
From bipartite cubic maps to simple maps



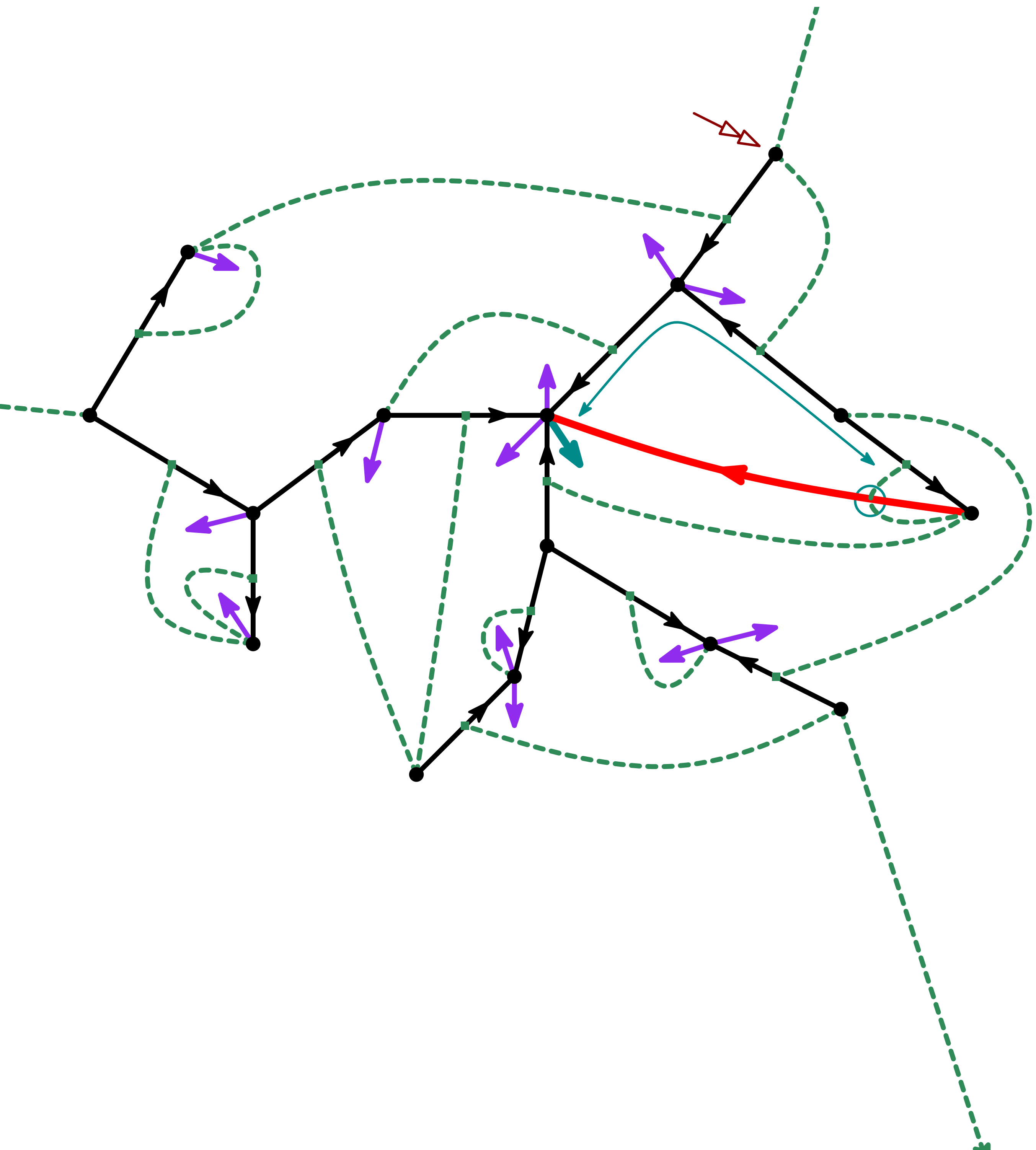
- Apply the following local rule :



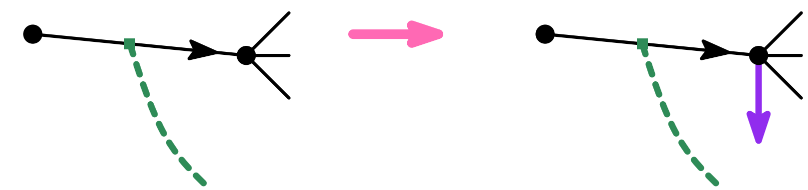
- Turning clockwise around the tree, do the following closures:



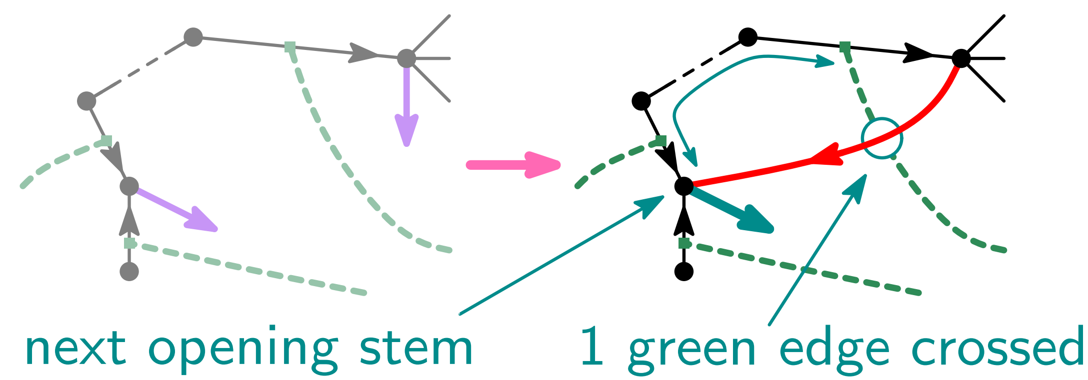
From bipartite cubic maps to simple maps



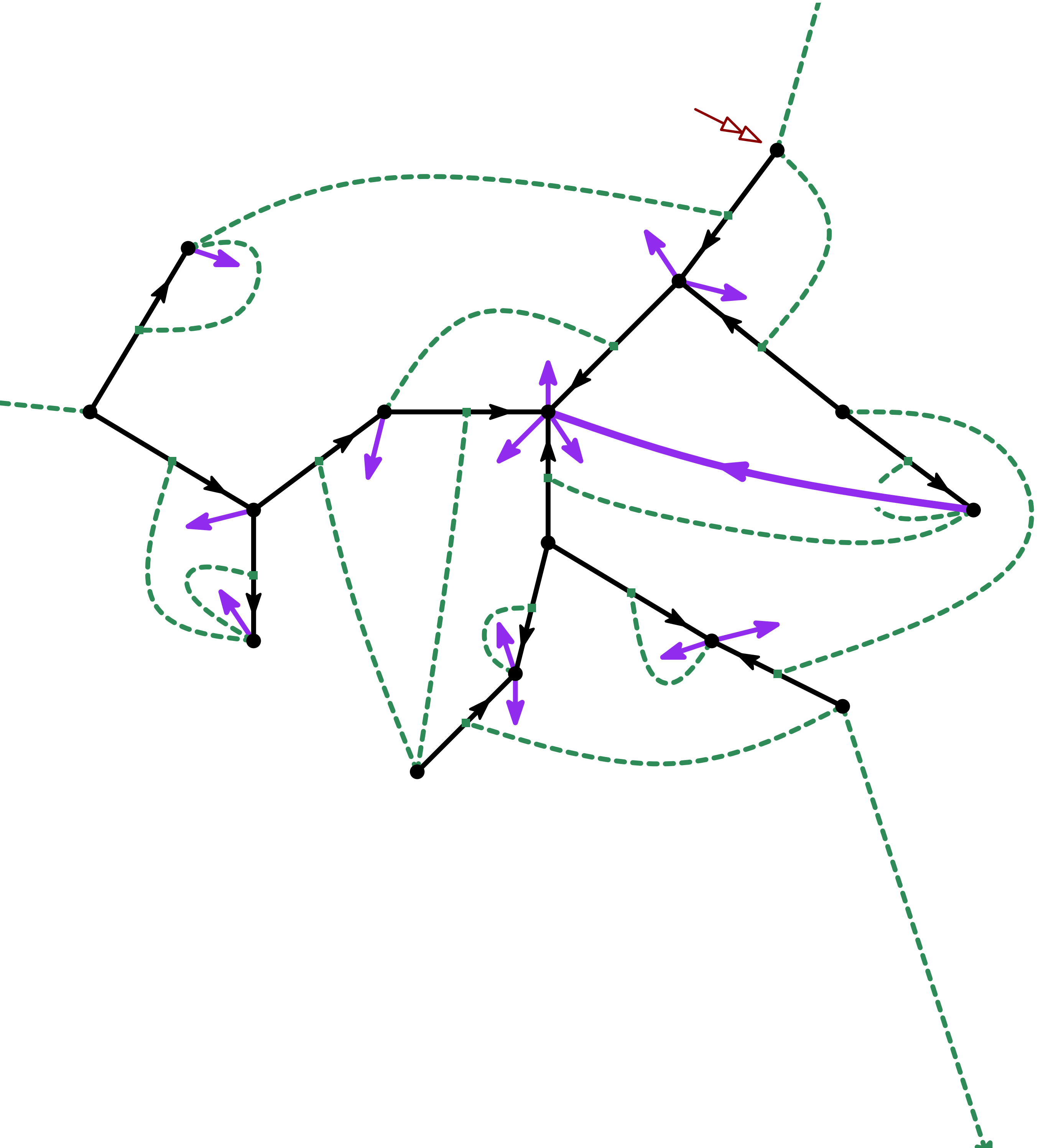
- Apply the following local rule :



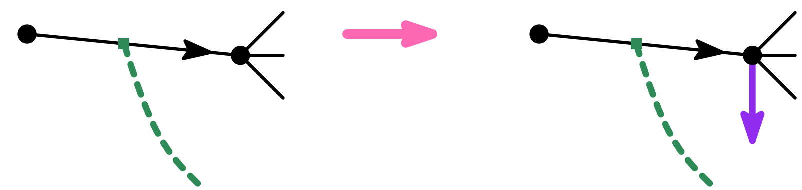
- Turning clockwise around the tree, do the following closures:



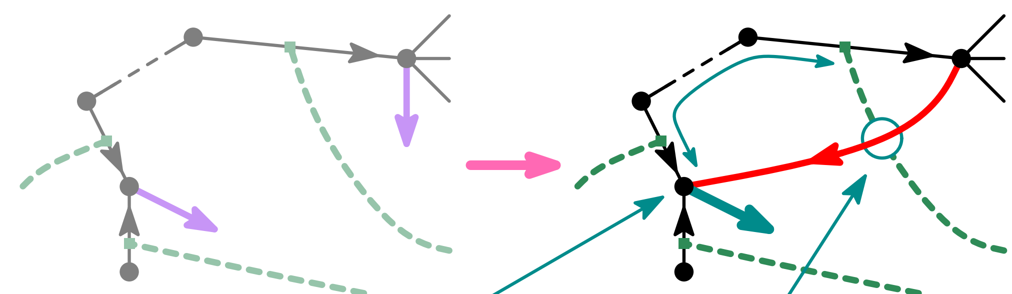
From bipartite cubic maps to simple maps



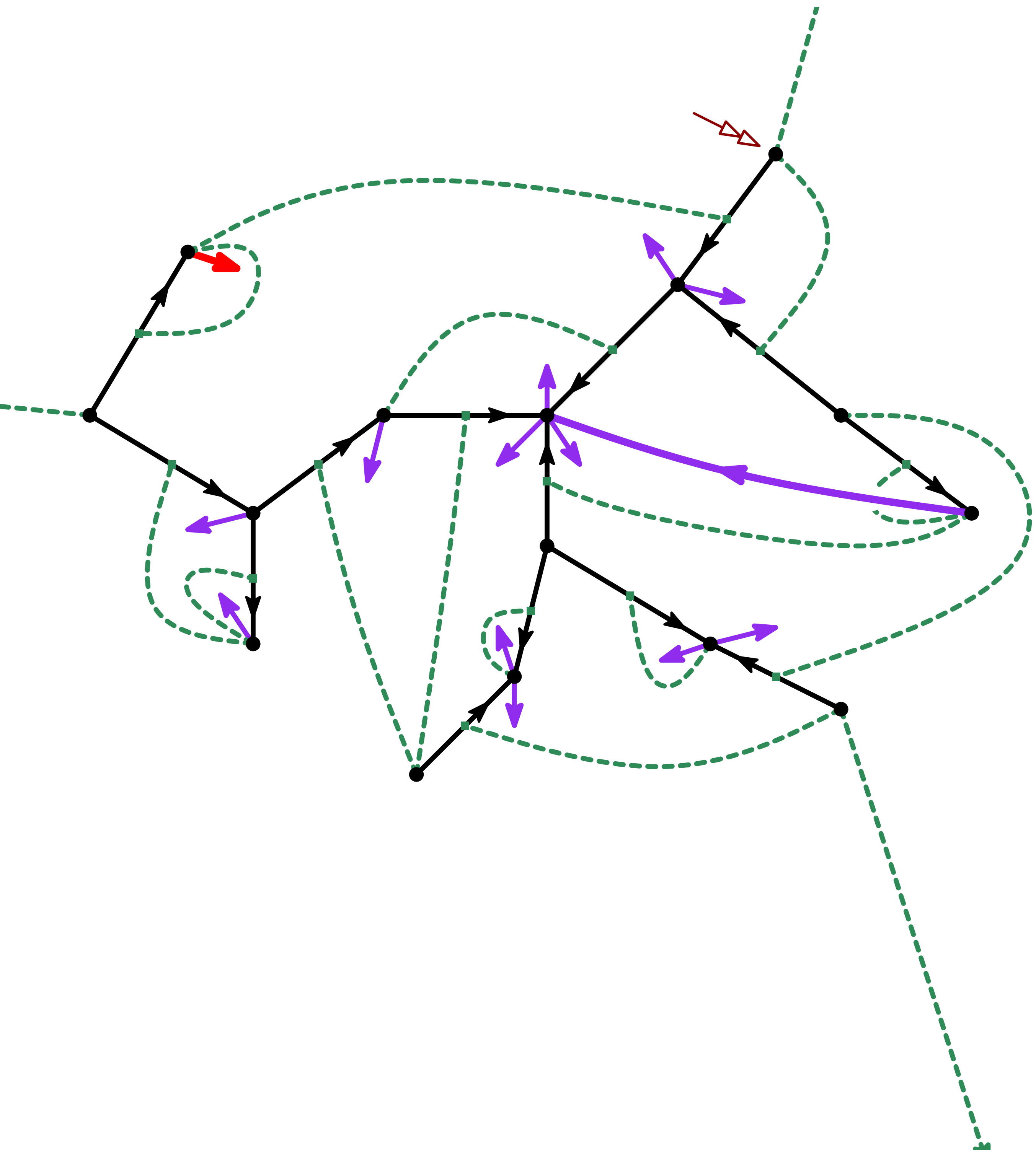
- Apply the following local rule :



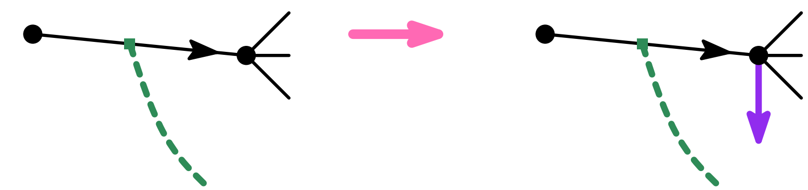
- Turning clockwise around the tree, do the following closures:



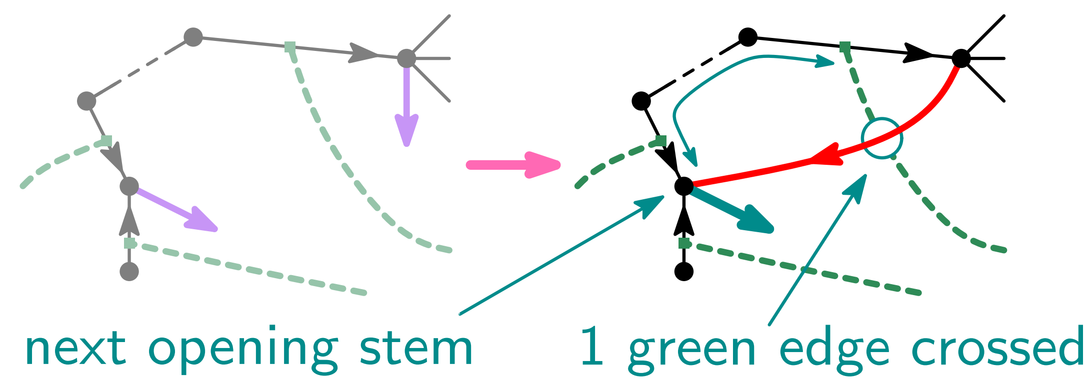
From bipartite cubic maps to simple maps



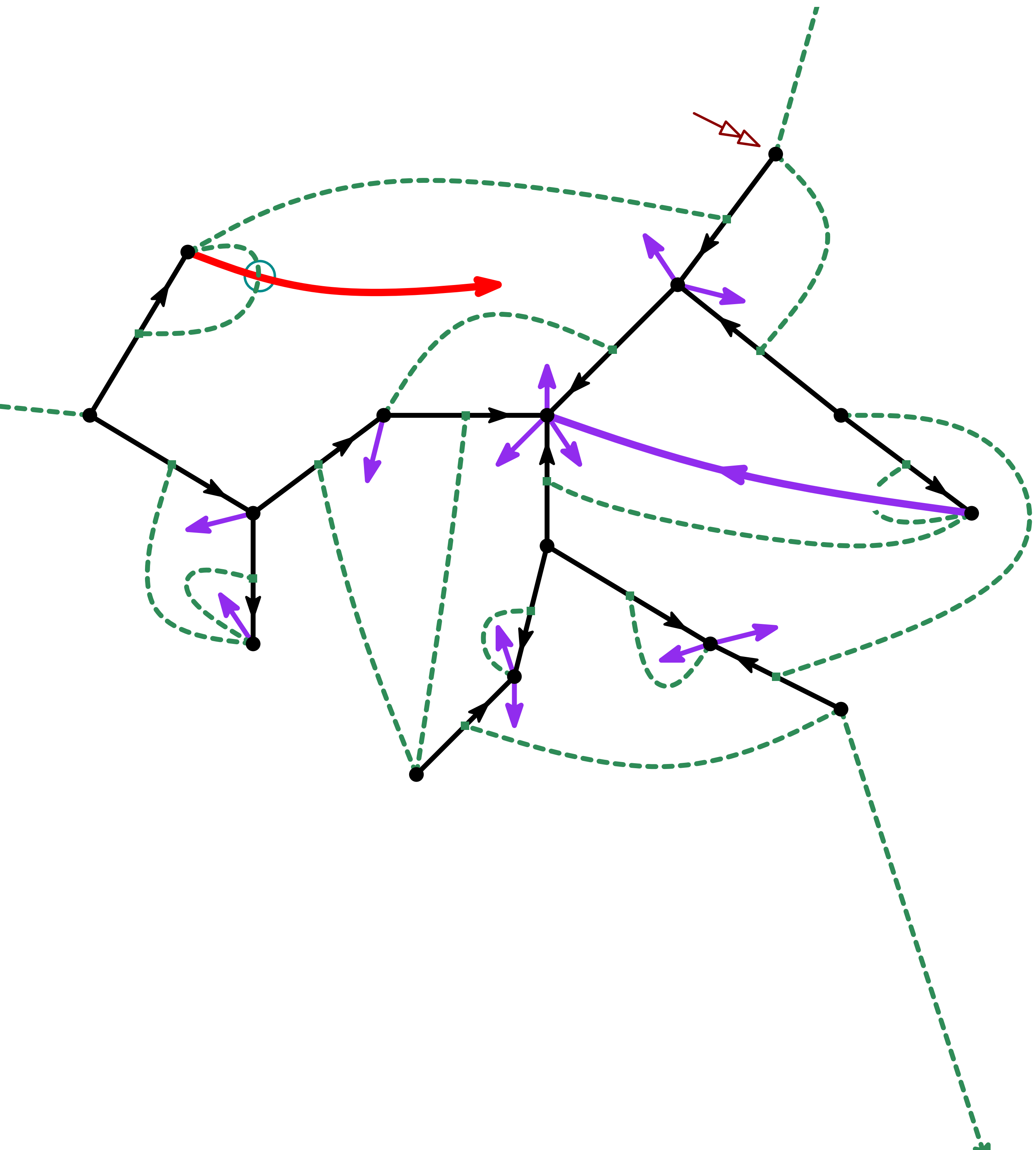
- Apply the following local rule :



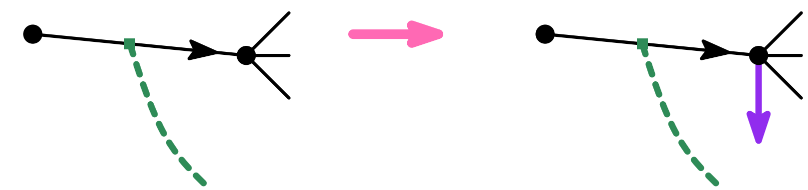
- Turning clockwise around the tree, do the following closures:



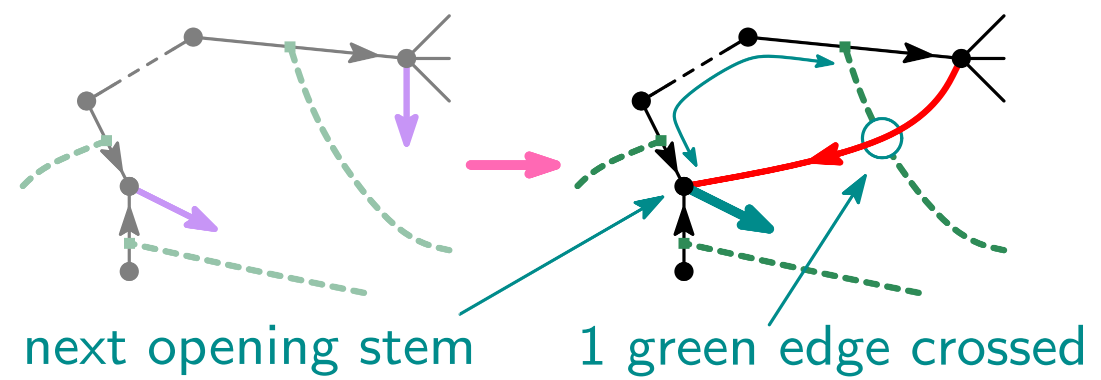
From bipartite cubic maps to simple maps



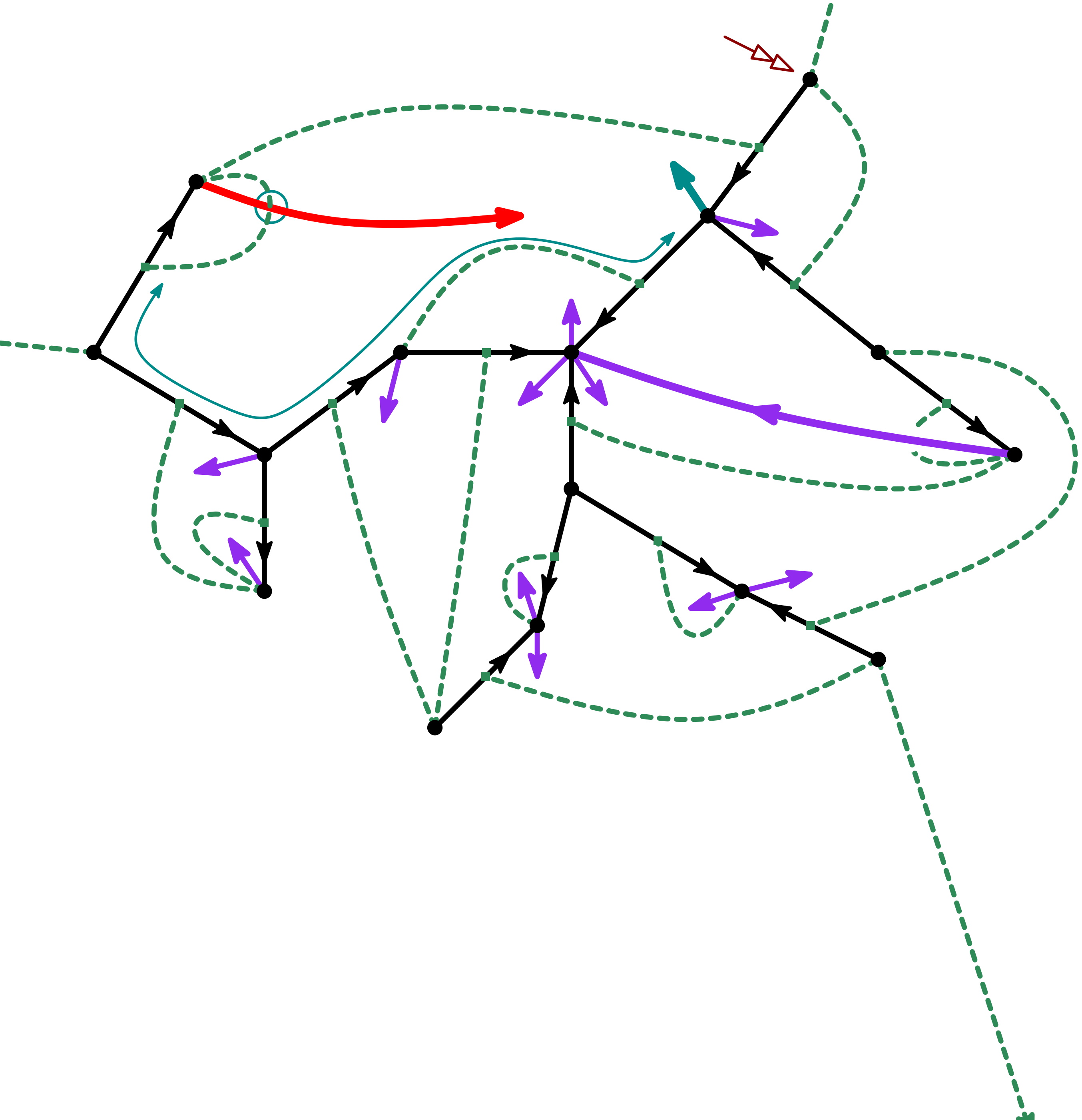
- Apply the following local rule :



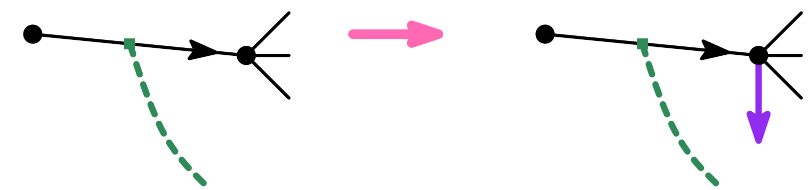
- Turning clockwise around the tree, do the following closures:



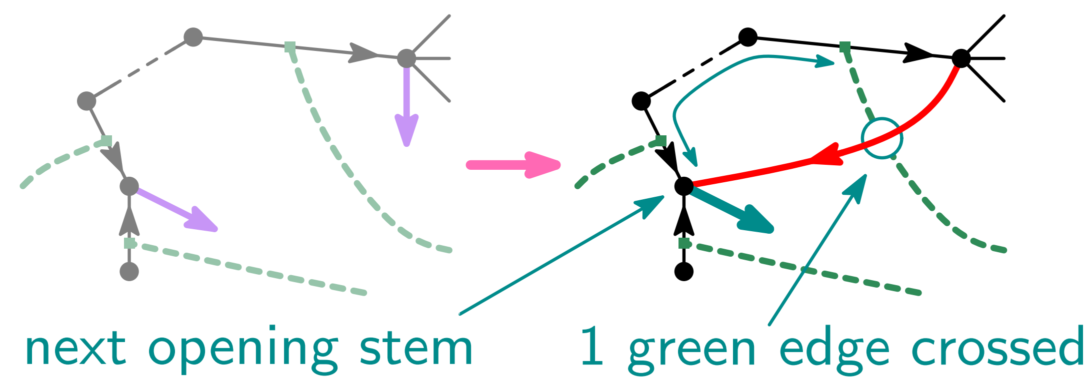
From bipartite cubic maps to simple maps



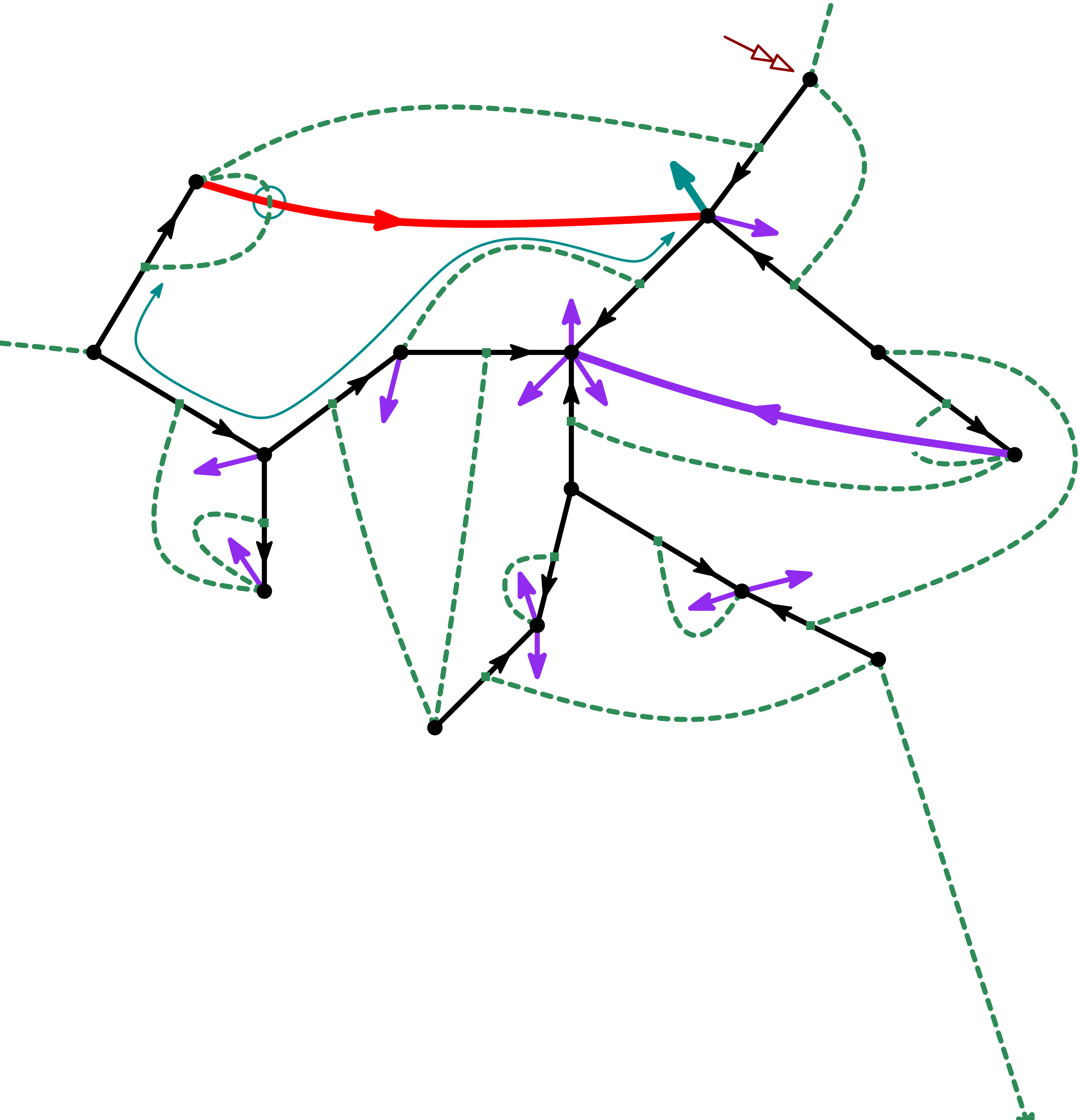
- Apply the following local rule :



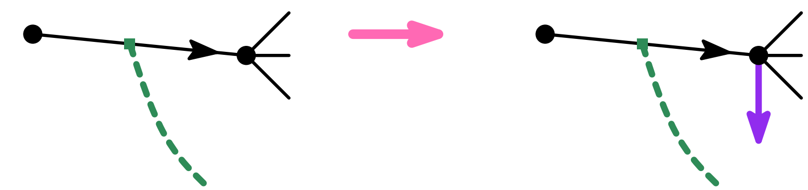
- Turning clockwise around the tree, do the following closures:



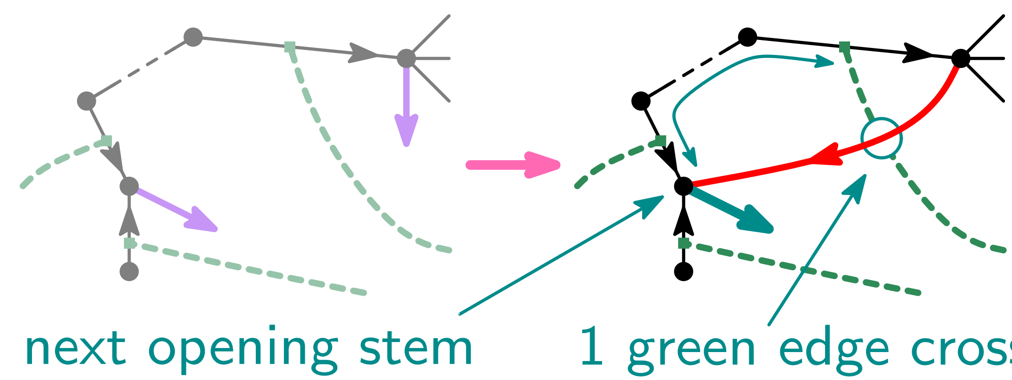
From bipartite cubic maps to simple maps



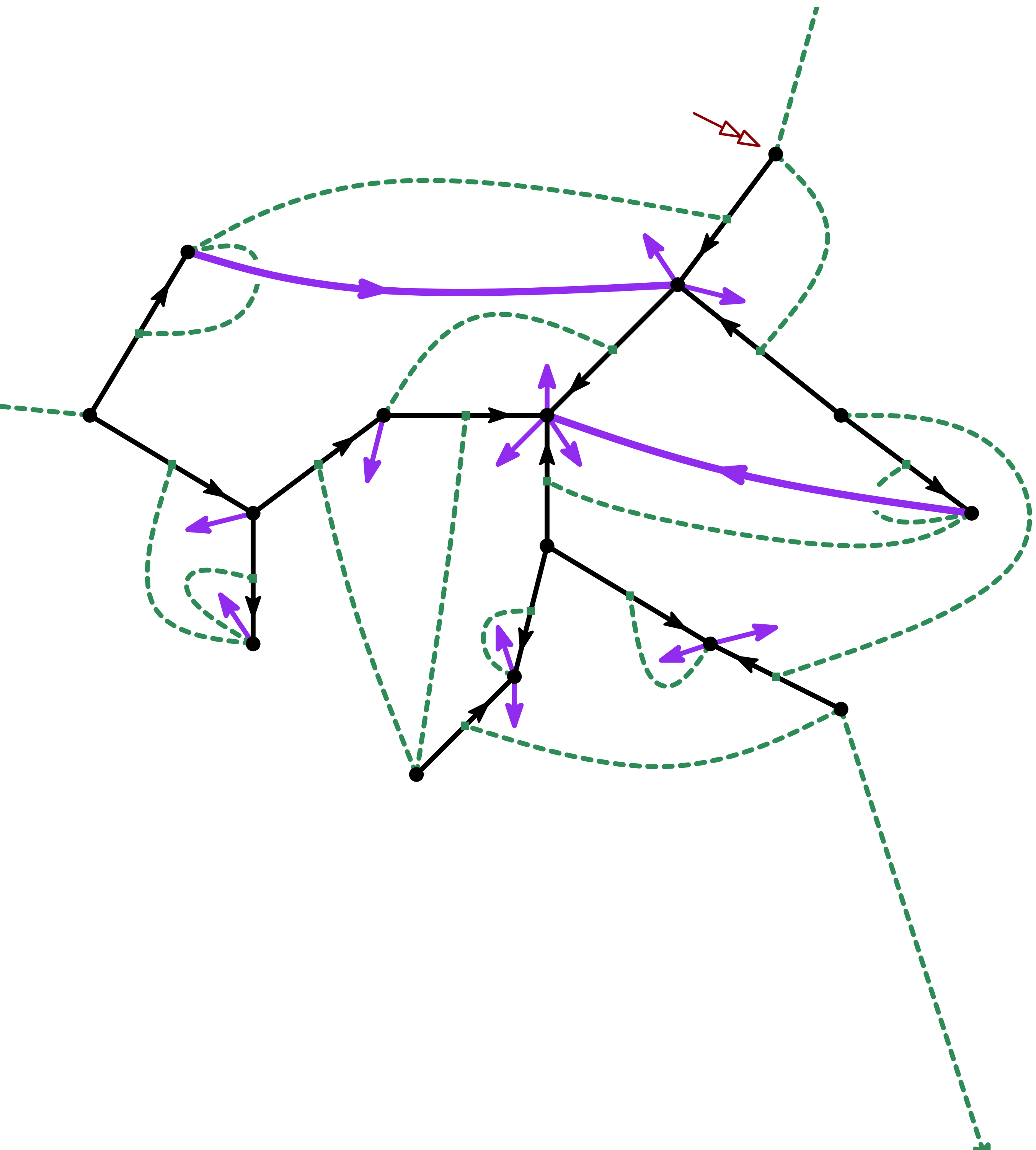
- Apply the following local rule :



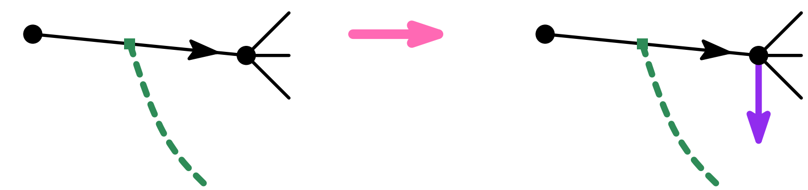
- Turning clockwise around the tree, do the following closures:



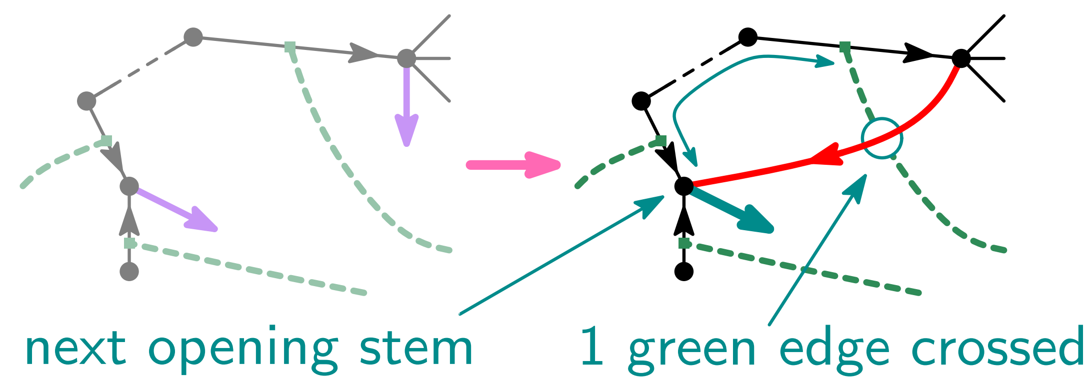
From bipartite cubic maps to simple maps



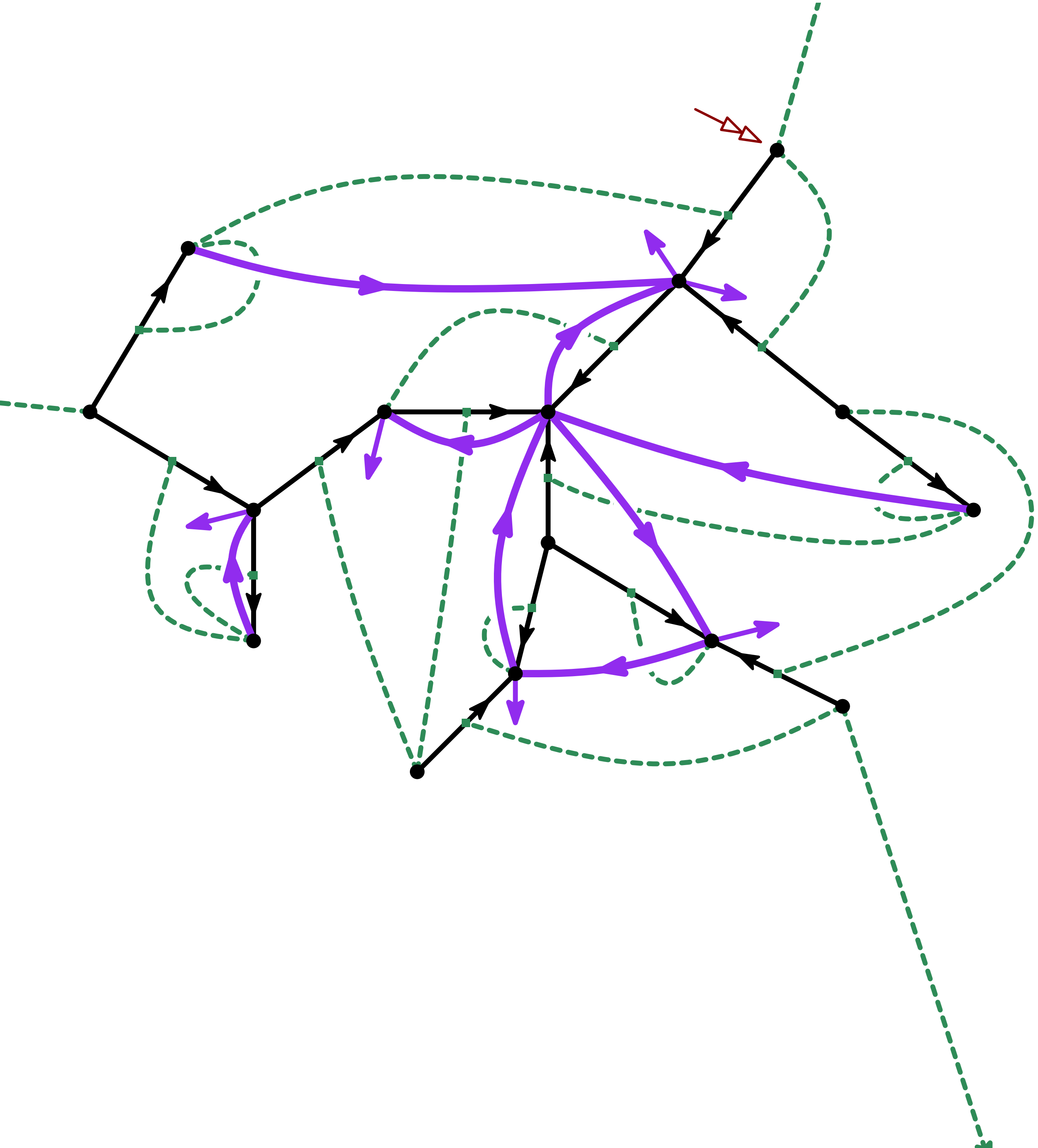
- Apply the following local rule :



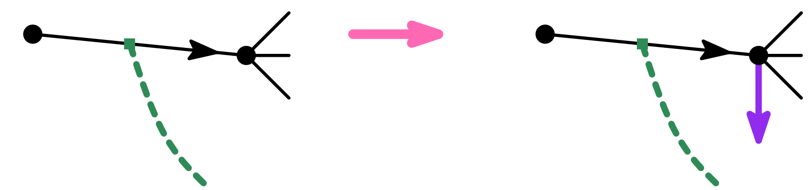
- Turning clockwise around the tree, do the following closures:



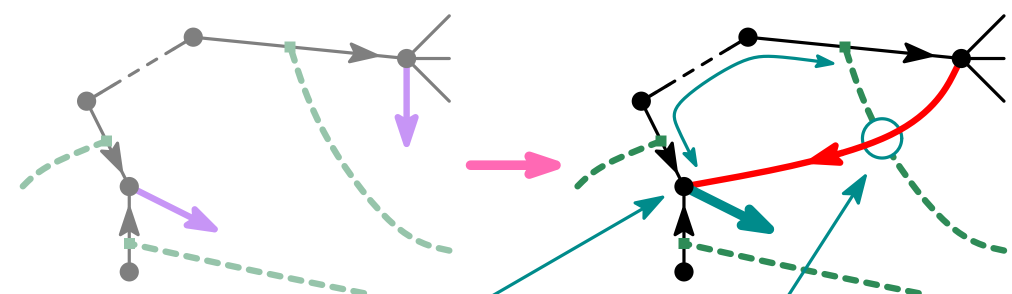
From bipartite cubic maps to simple maps



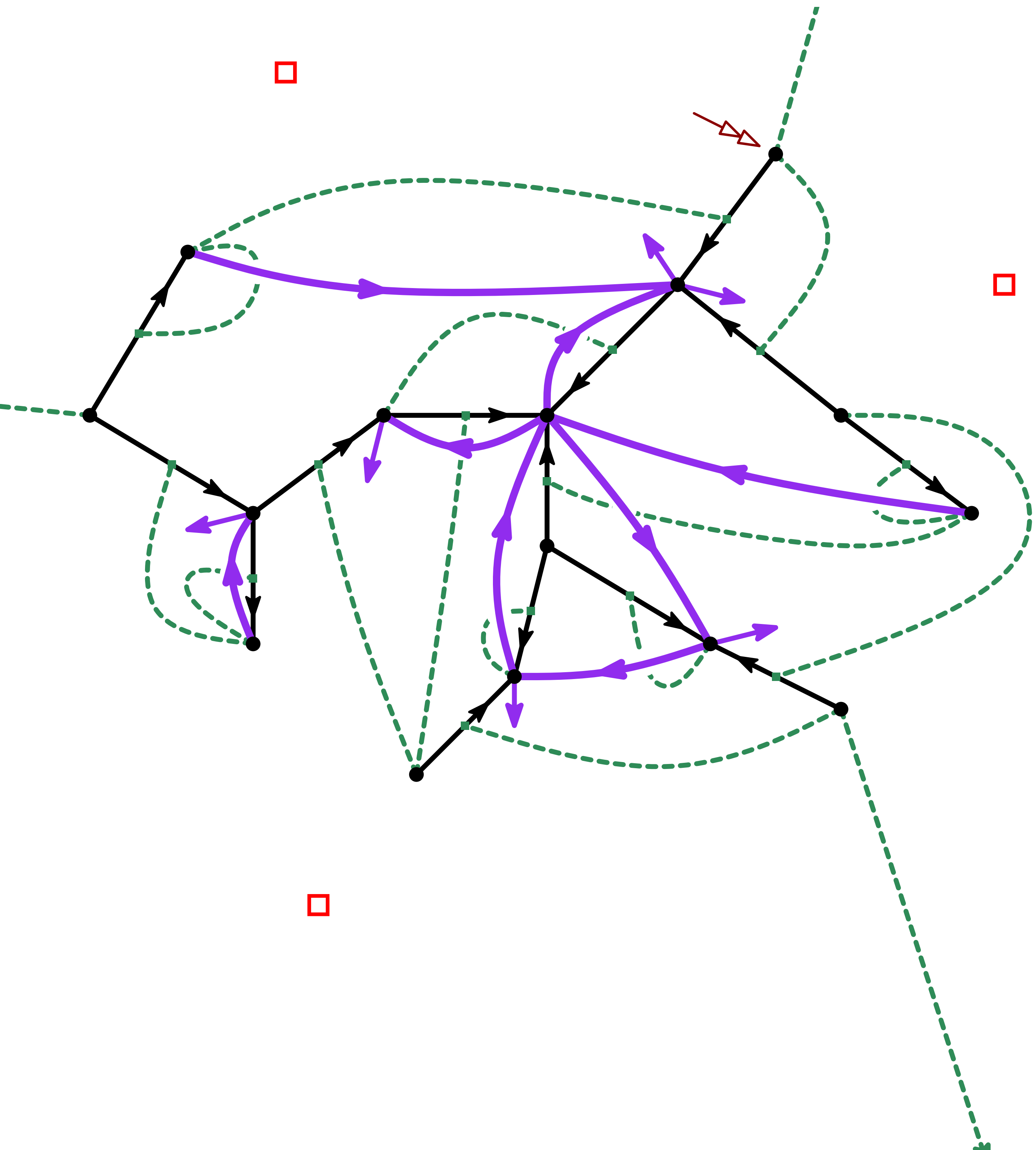
- Apply the following local rule :



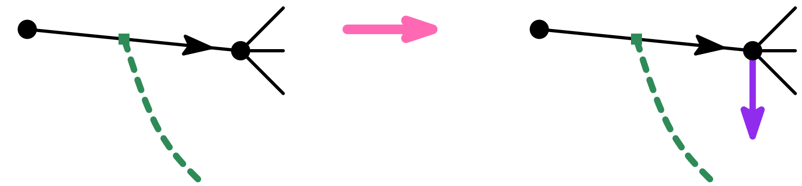
- Turning clockwise around the tree, do the following closures:



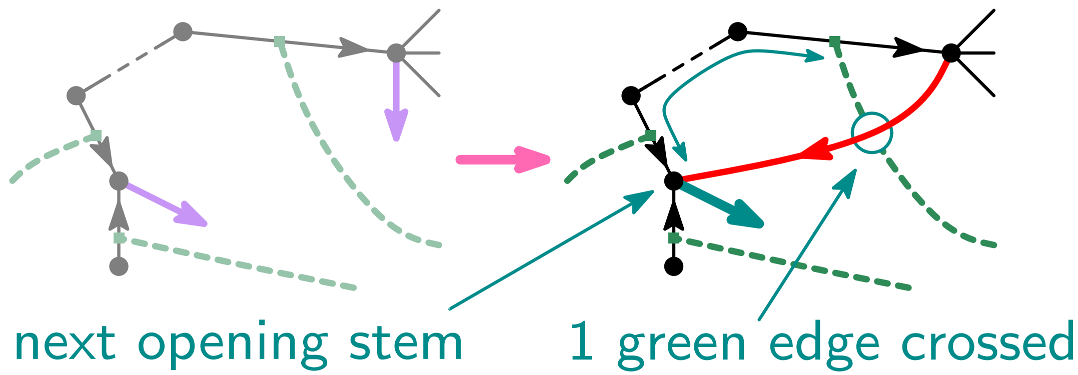
From bipartite cubic maps to simple maps



- Apply the following local rule :

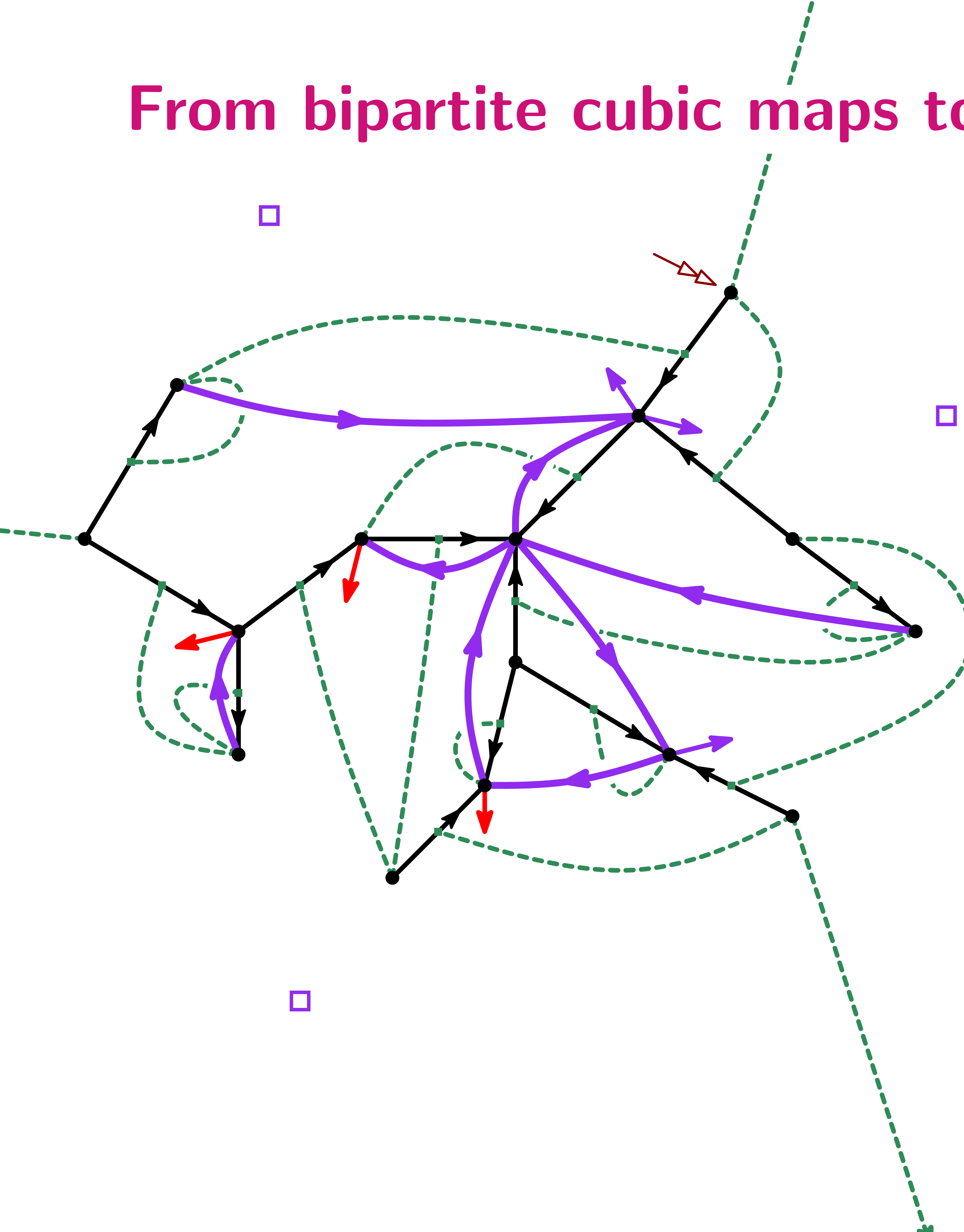


- Turning clockwise around the tree, do the following closures:

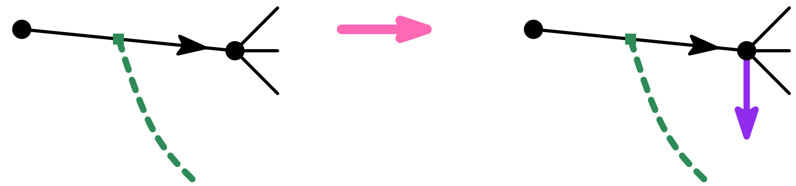


- Add 3 vertices and close the remaining opening stems sector by sector

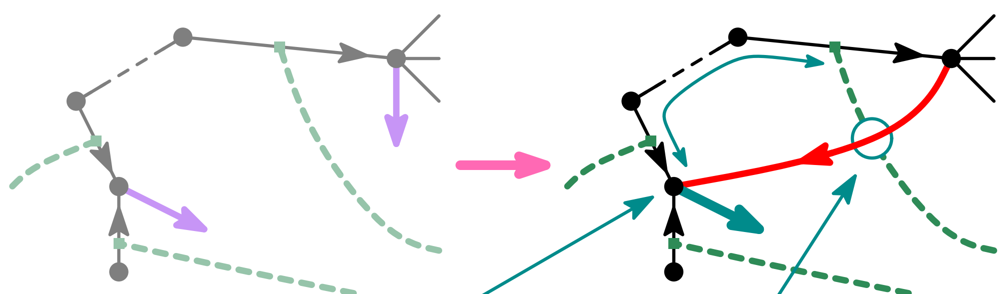
From bipartite cubic maps to simple maps



- Apply the following local rule :

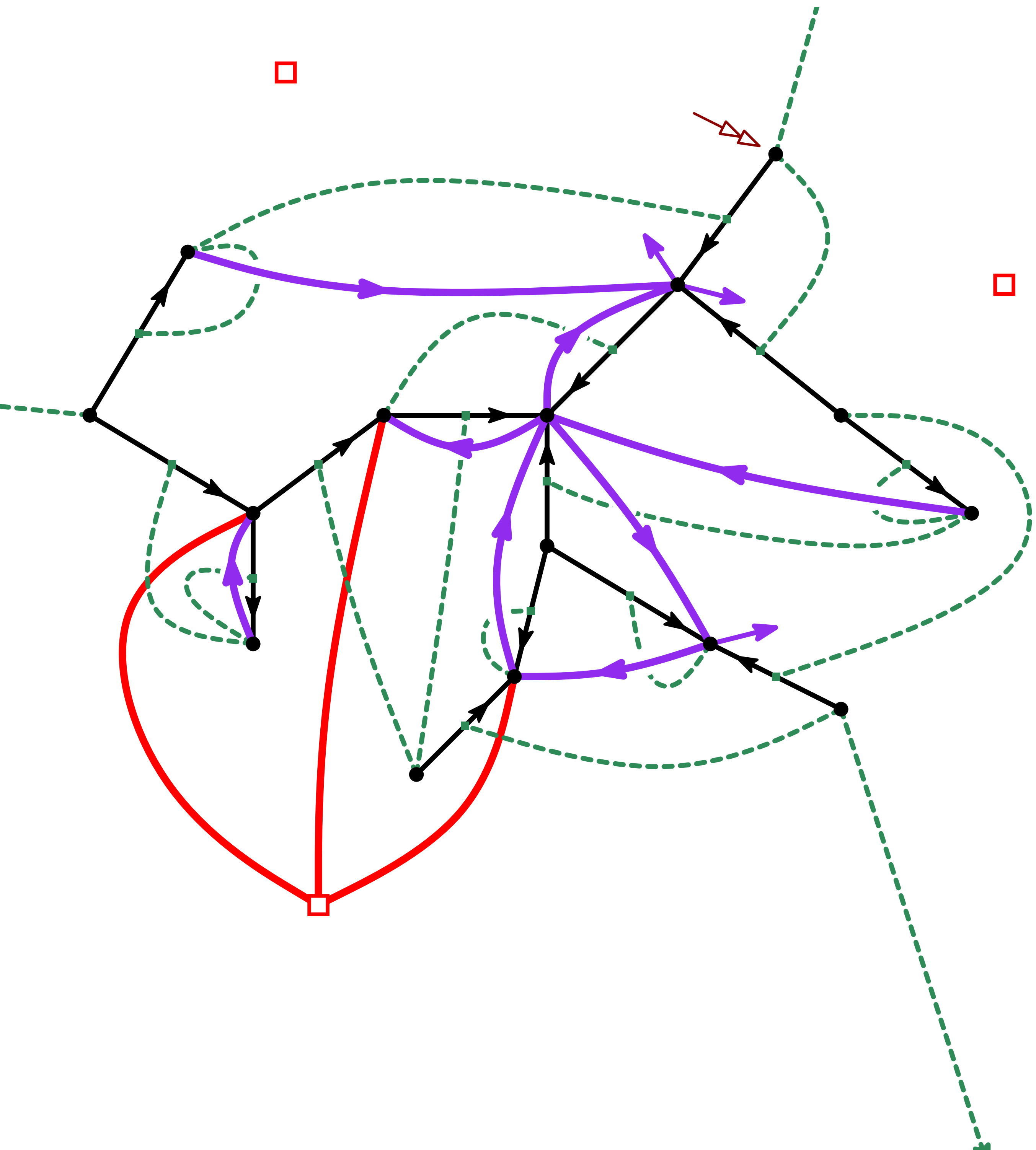


- Turning clockwise around the tree, do the following closures:

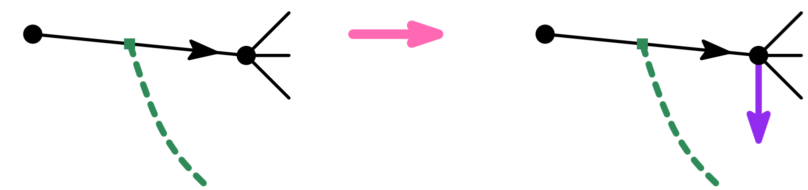


- Add 3 vertices and close the remaining opening stems sector by sector

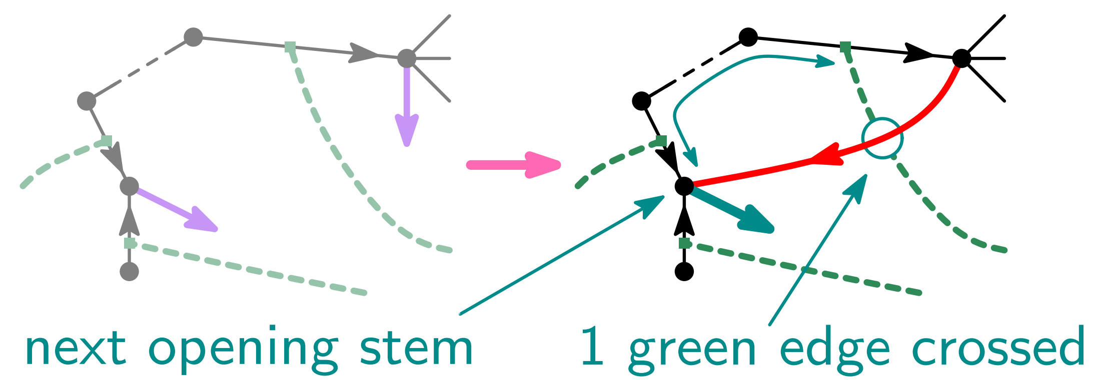
From bipartite cubic maps to simple maps



- Apply the following local rule :

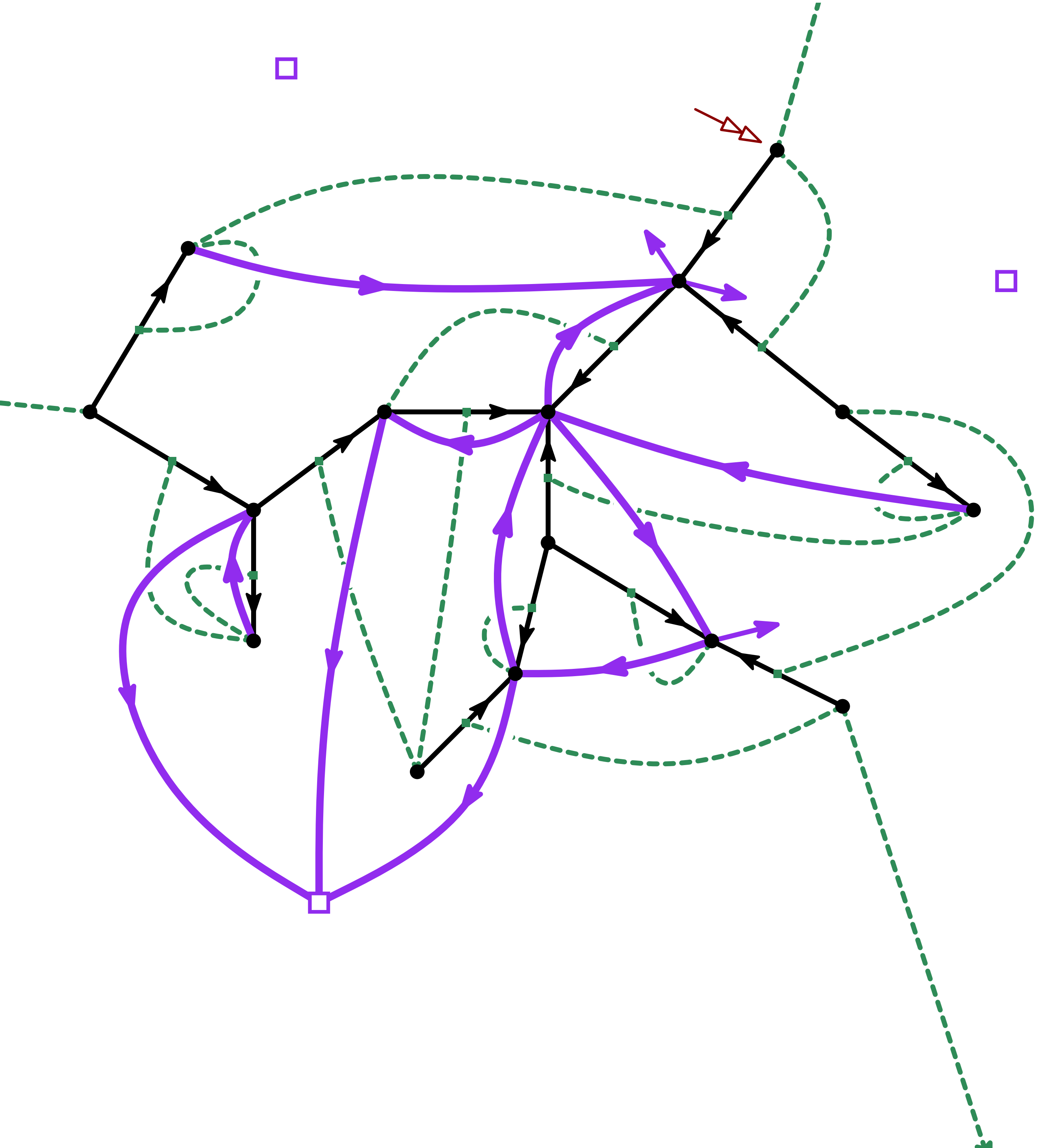


- Turning clockwise around the tree, do the following closures:

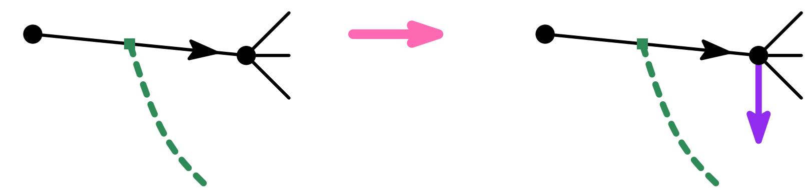


- Add 3 vertices and close the remaining opening stems sector by sector

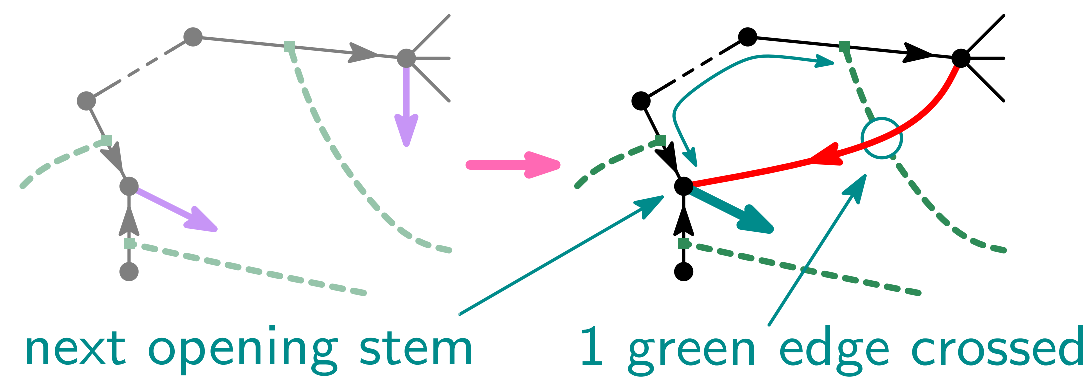
From bipartite cubic maps to simple maps



- Apply the following local rule :

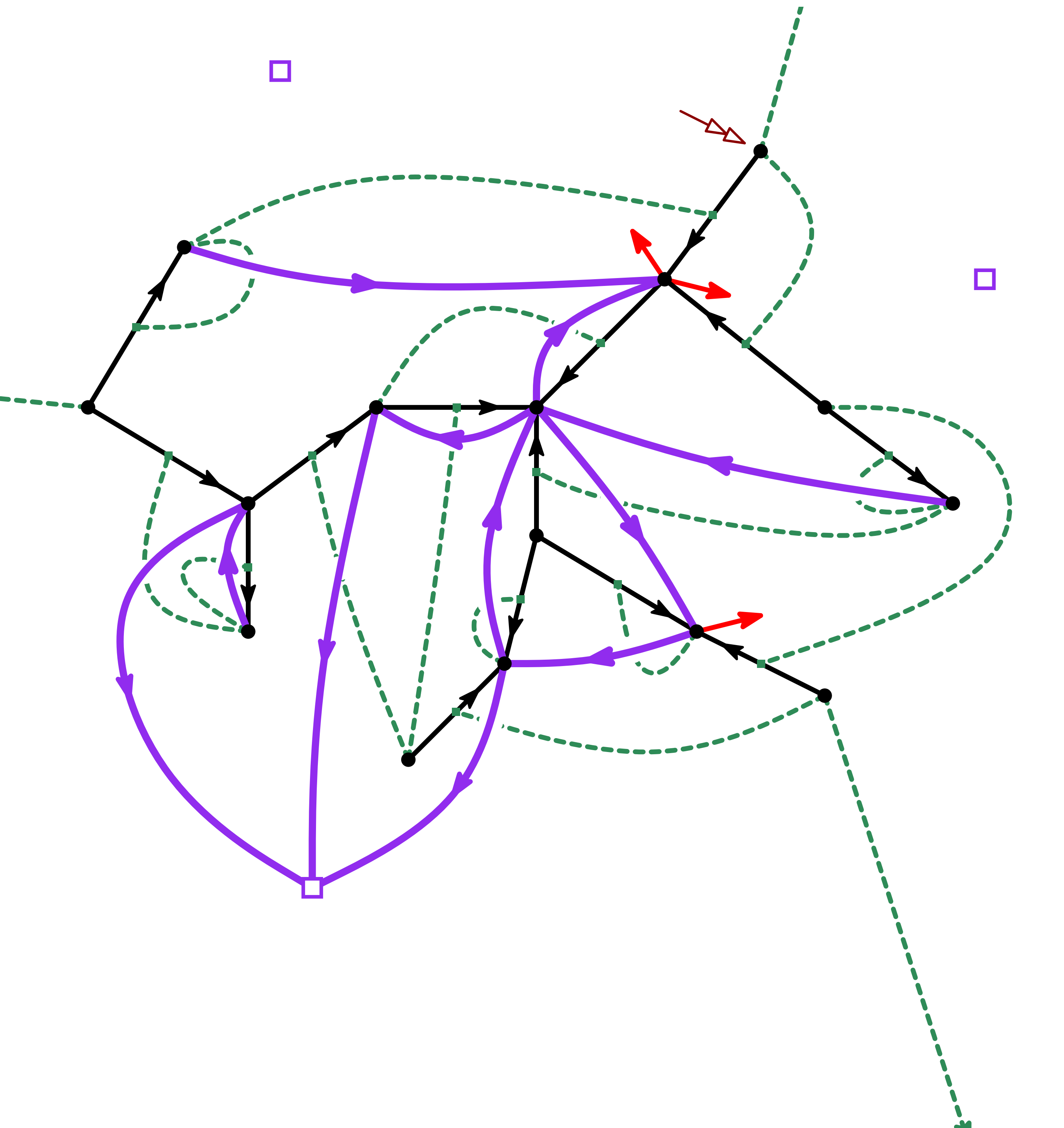


- Turning clockwise around the tree, do the following closures:

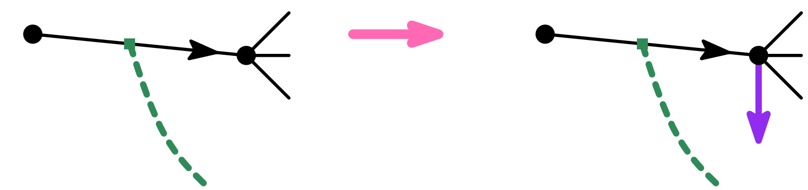


- Add 3 vertices and close the remaining opening stems sector by sector

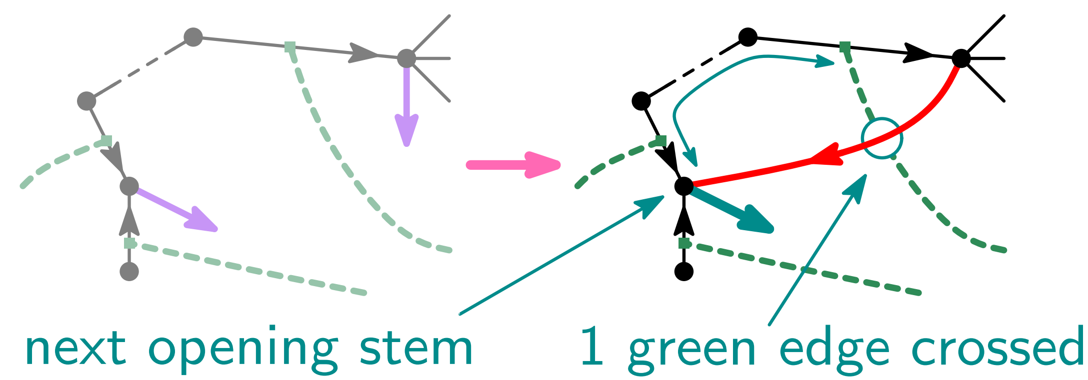
From bipartite cubic maps to simple maps



- Apply the following local rule :

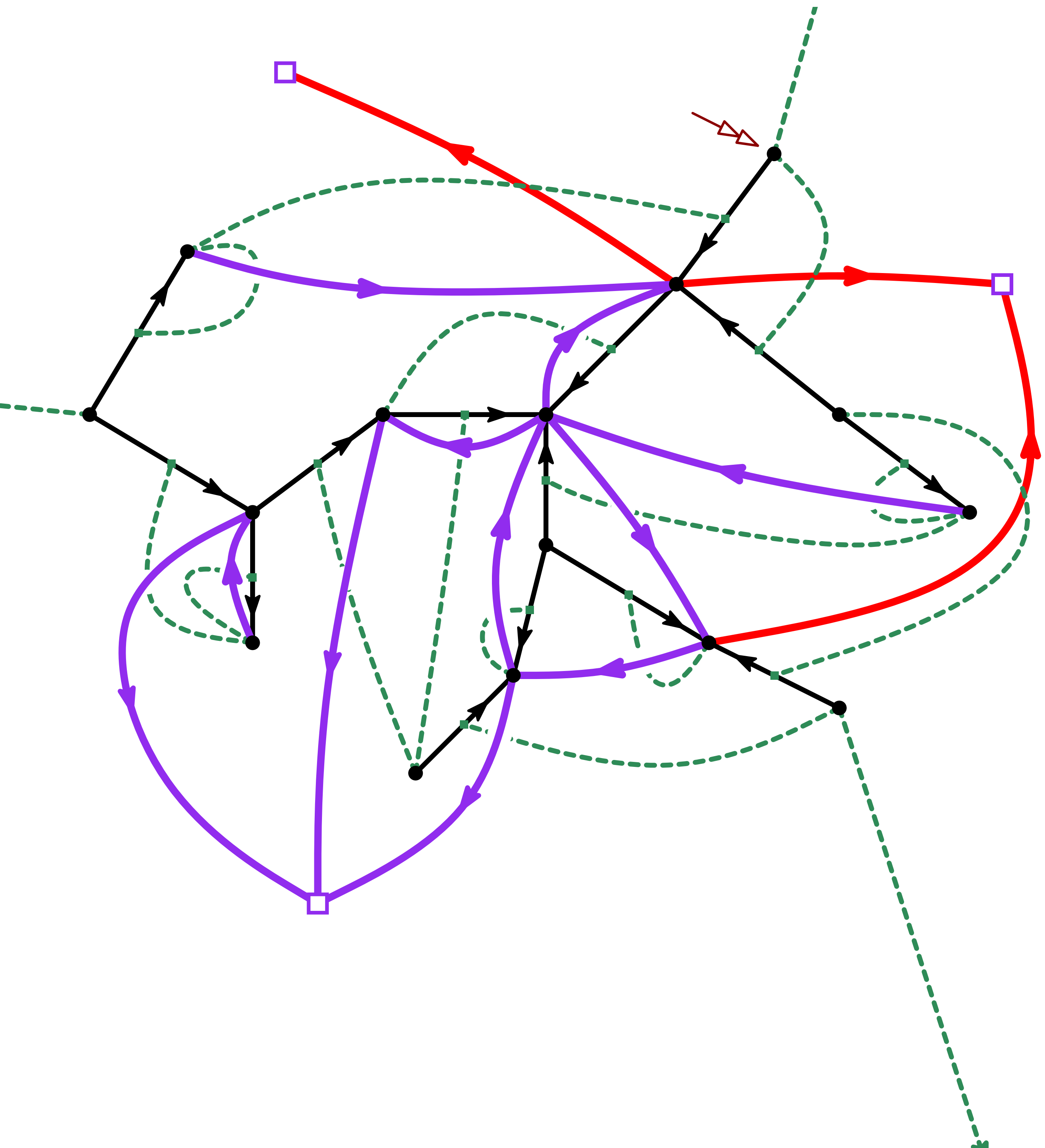


- Turning clockwise around the tree, do the following closures:

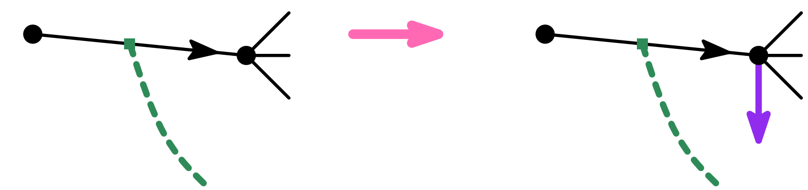


- Add 3 vertices and close the remaining opening stems sector by sector

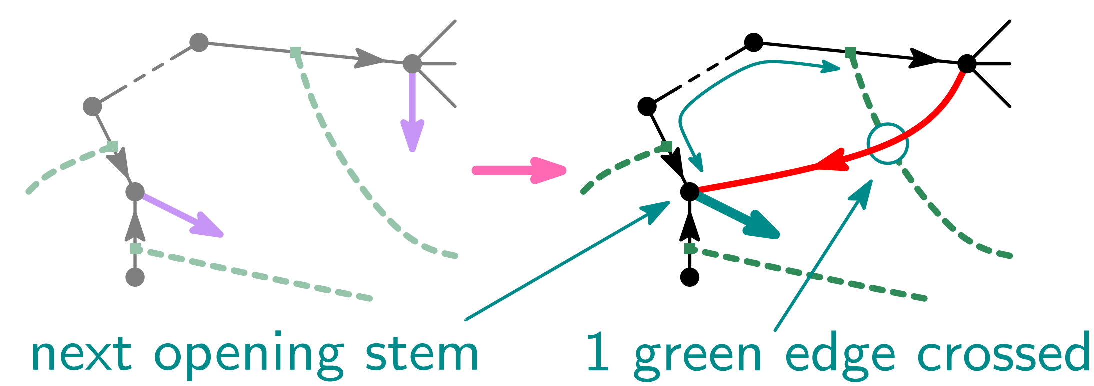
From bipartite cubic maps to simple maps



- Apply the following local rule :

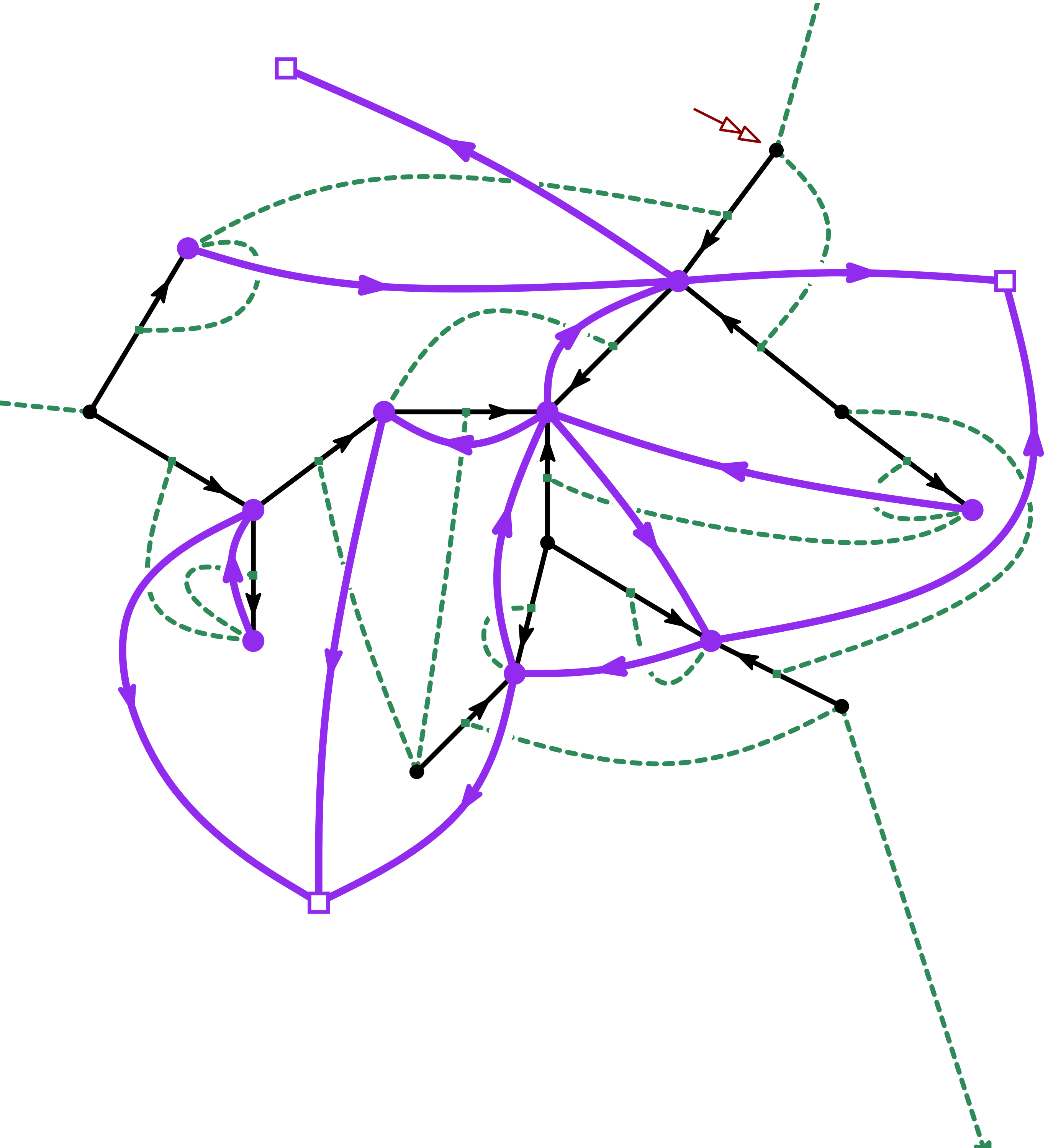


- Turning clockwise around the tree, do the following closures:

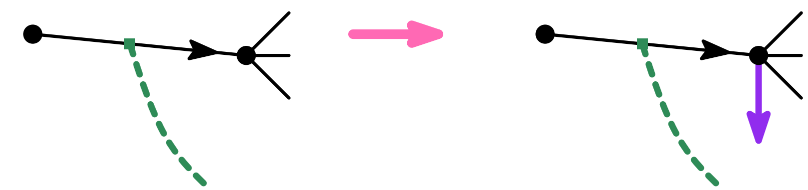


- Add 3 vertices and close the remaining opening stems sector by sector

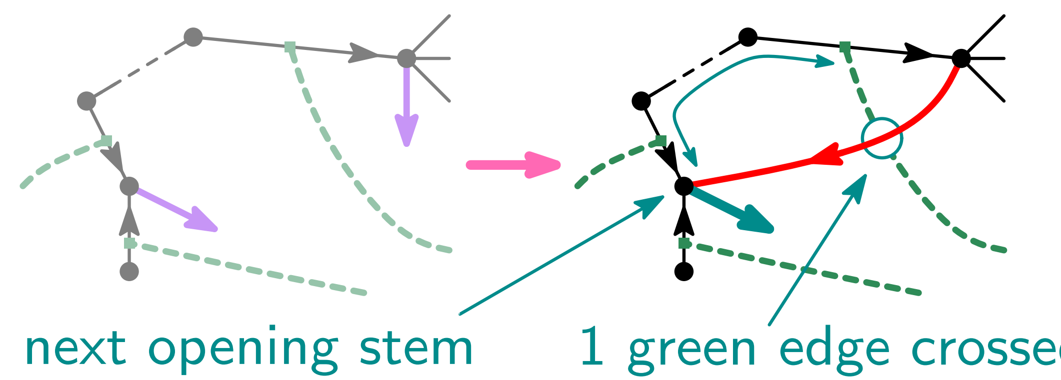
From bipartite cubic maps to simple maps



- Apply the following local rule :

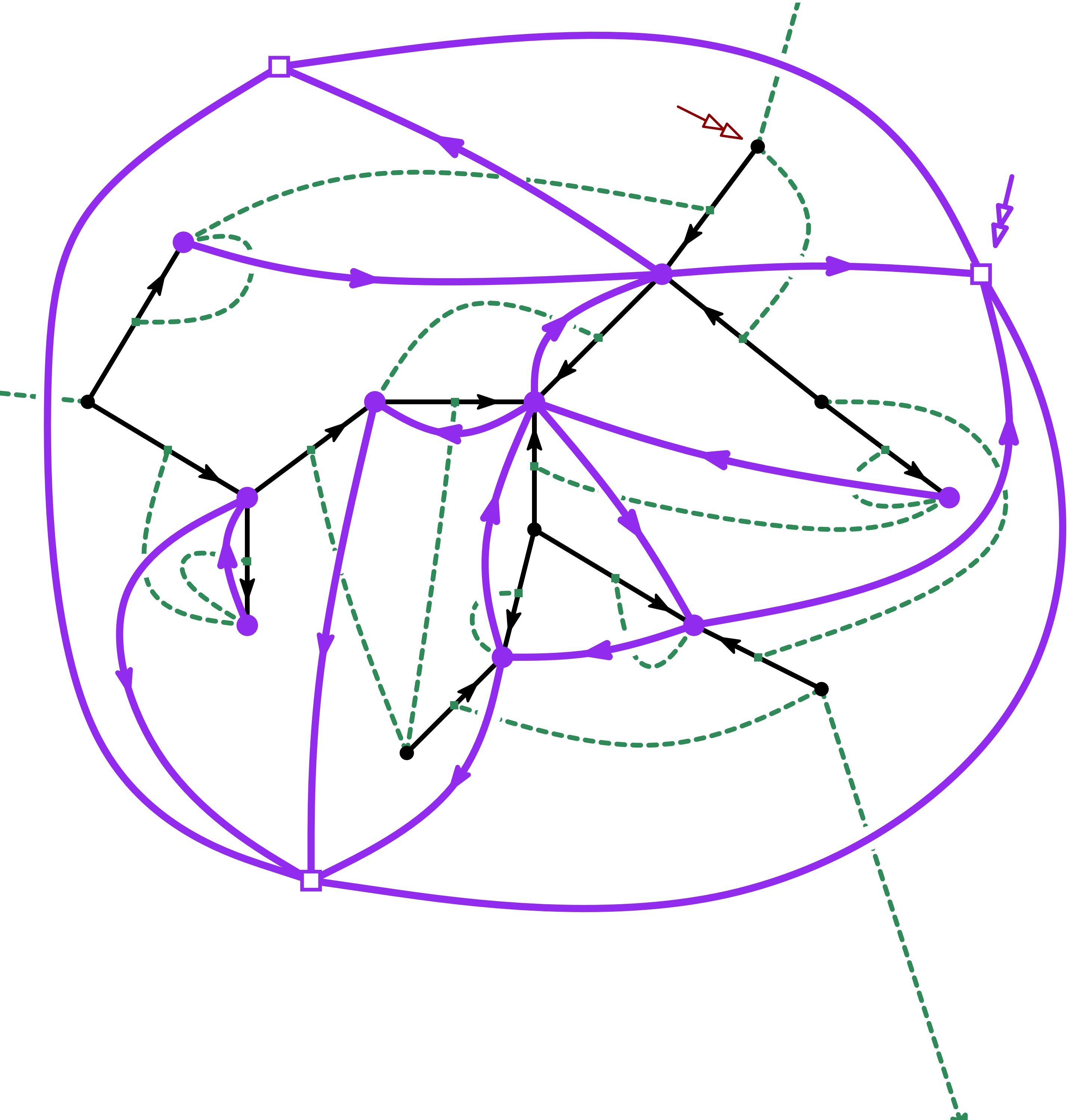


- Turning clockwise around the tree, do the following closures:

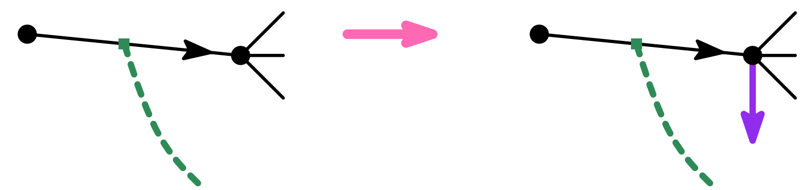


- Add 3 vertices and close the remaining opening stems sector by sector

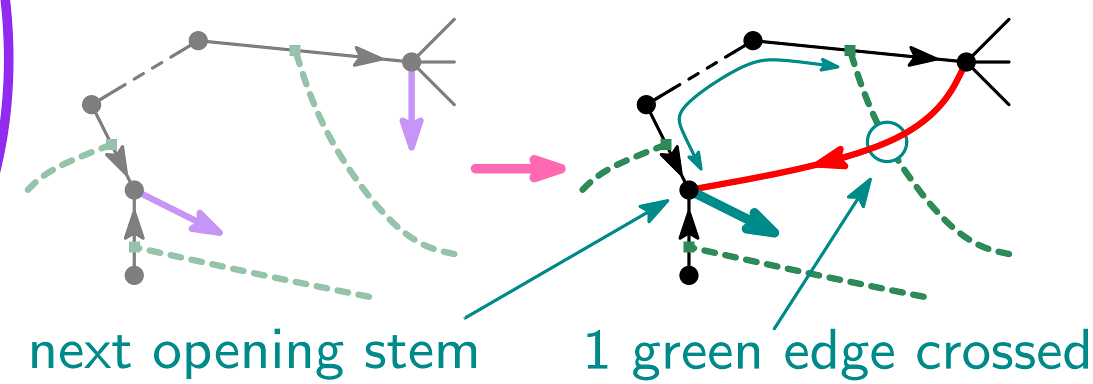
From bipartite cubic maps to simple maps



- Apply the following local rule :

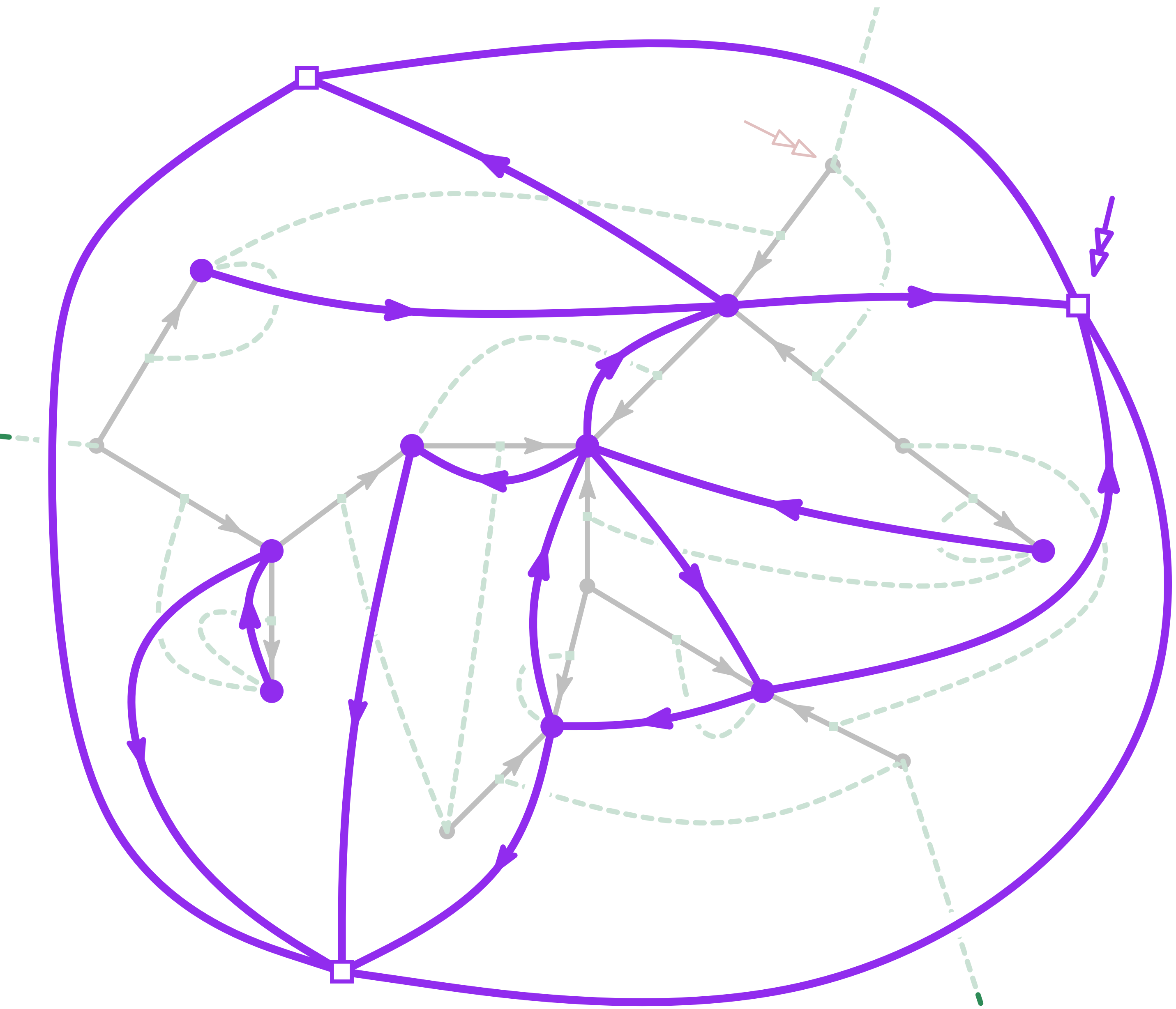


- Turning clockwise around the tree, do the following closures:

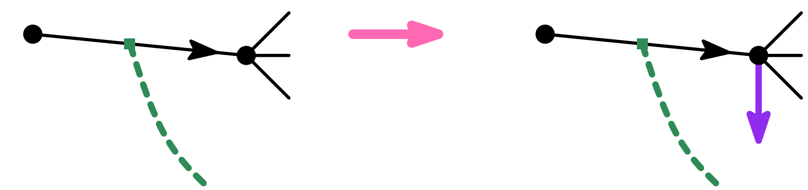


- Add 3 vertices and close the remaining opening stems sector by sector
- Connect the 3 outer vertices into a triangle

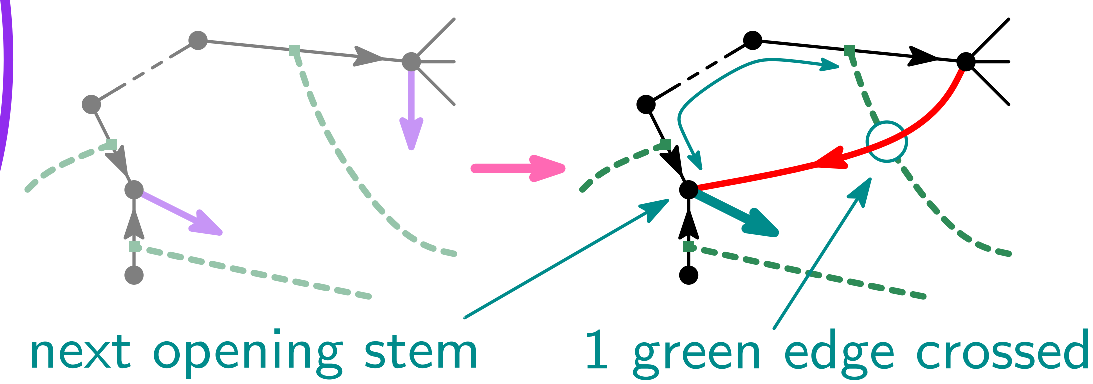
From bipartite cubic maps to simple maps



- Apply the following local rule :



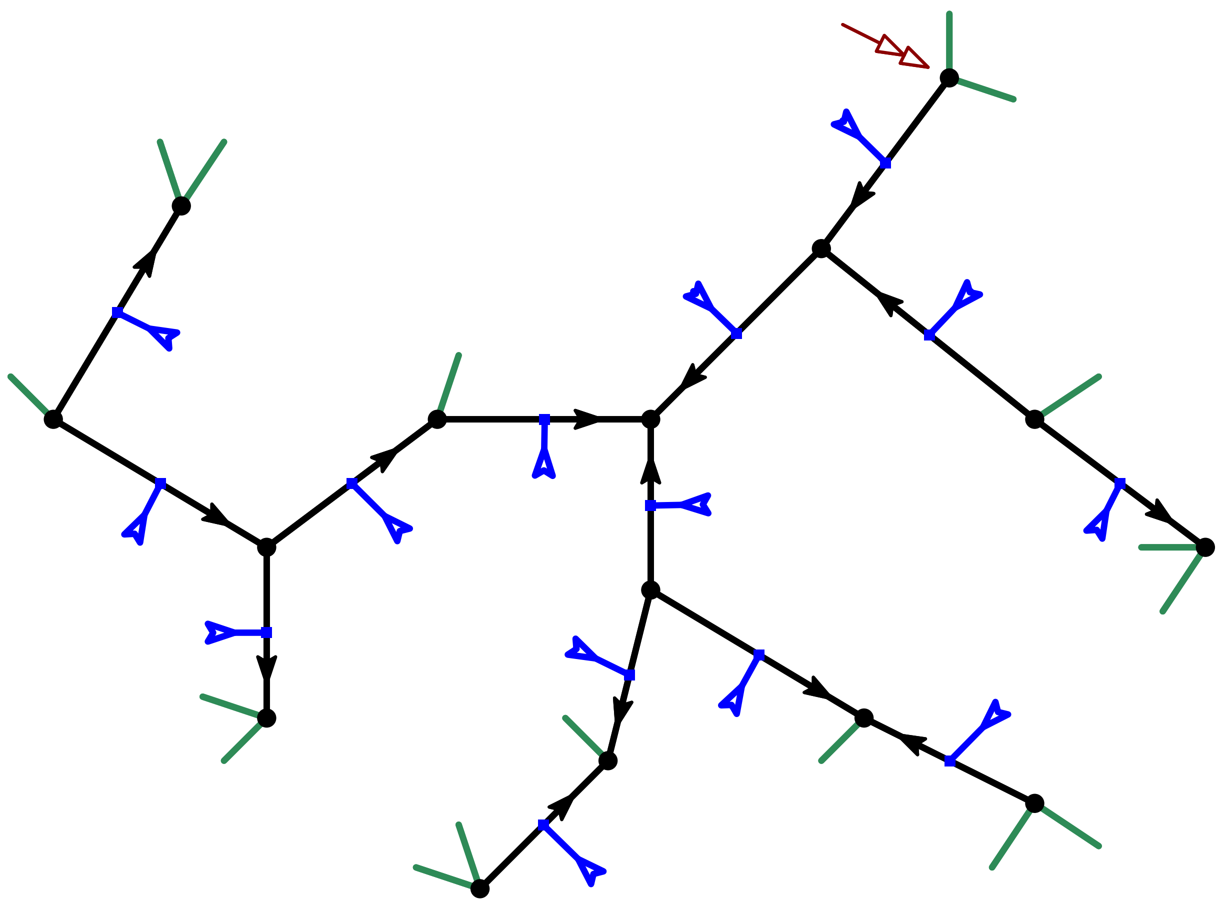
- Turning clockwise around the tree, do the following closures:



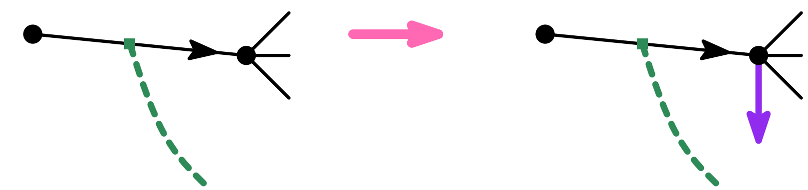
- Add 3 vertices and close the remaining opening stems sector by sector

Theorem : [ABCF] This is a bijection between **outer-triangular simple maps** and **bipartite cubic maps**

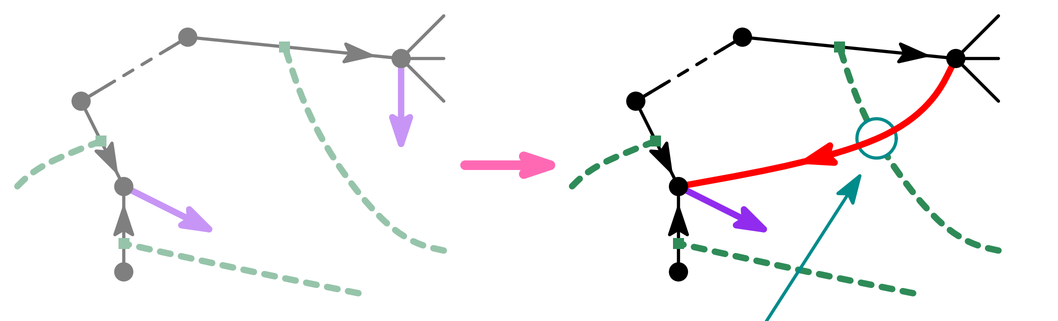
From bipartite cubic maps to simple maps



- Apply the following local rule :

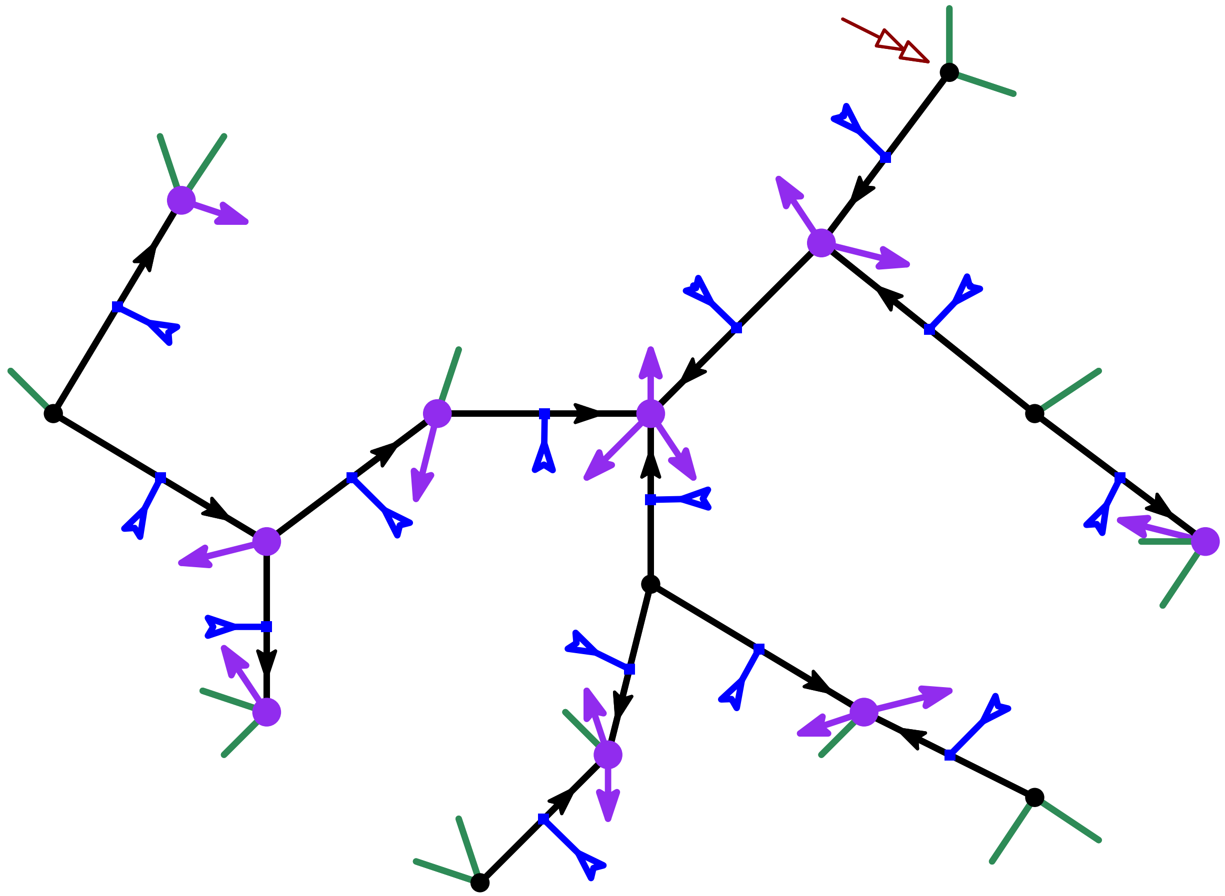


- Turning clockwise around the tree, do the following closures:

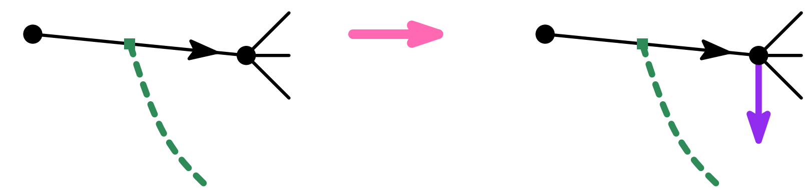


Theorem : [ABCF] This is a bijection between **outer-triangular simple maps** and **bipartite cubic maps**

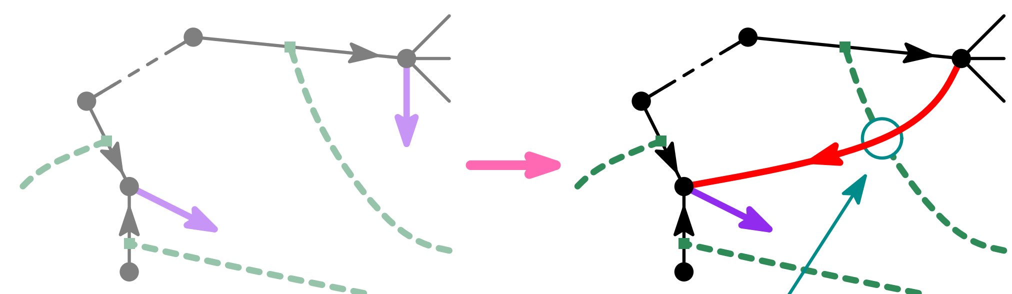
From bipartite cubic maps to simple maps



- Apply the following local rule :

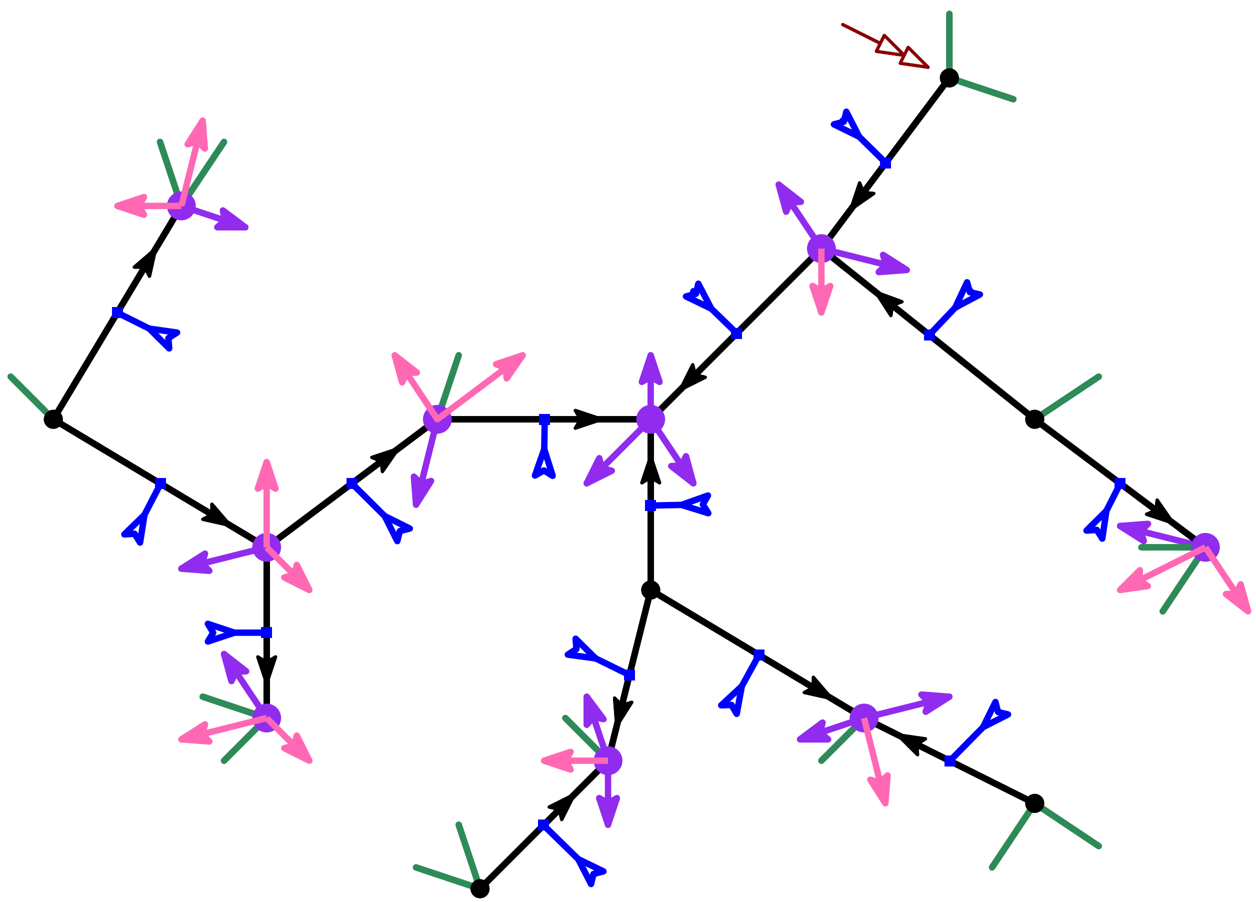


- Turning clockwise around the tree, do the following closures:

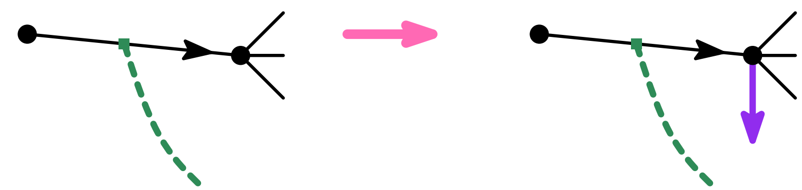


Theorem : [ABCF] This is a bijection between **outer-triangular simple maps** and **bipartite cubic maps**

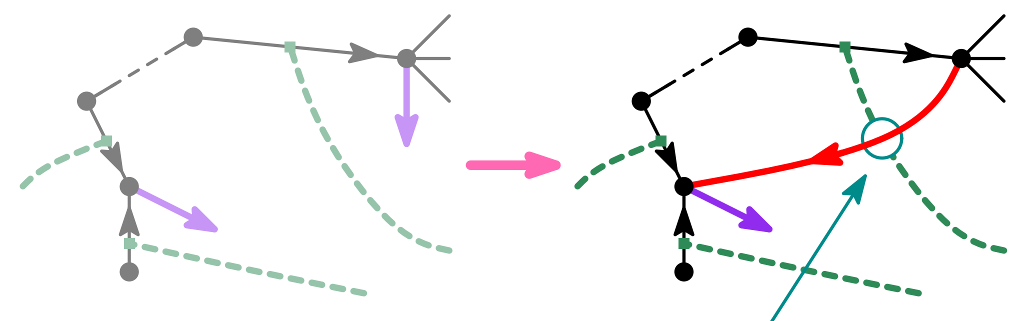
From bipartite cubic maps to simple maps



- Apply the following local rule :

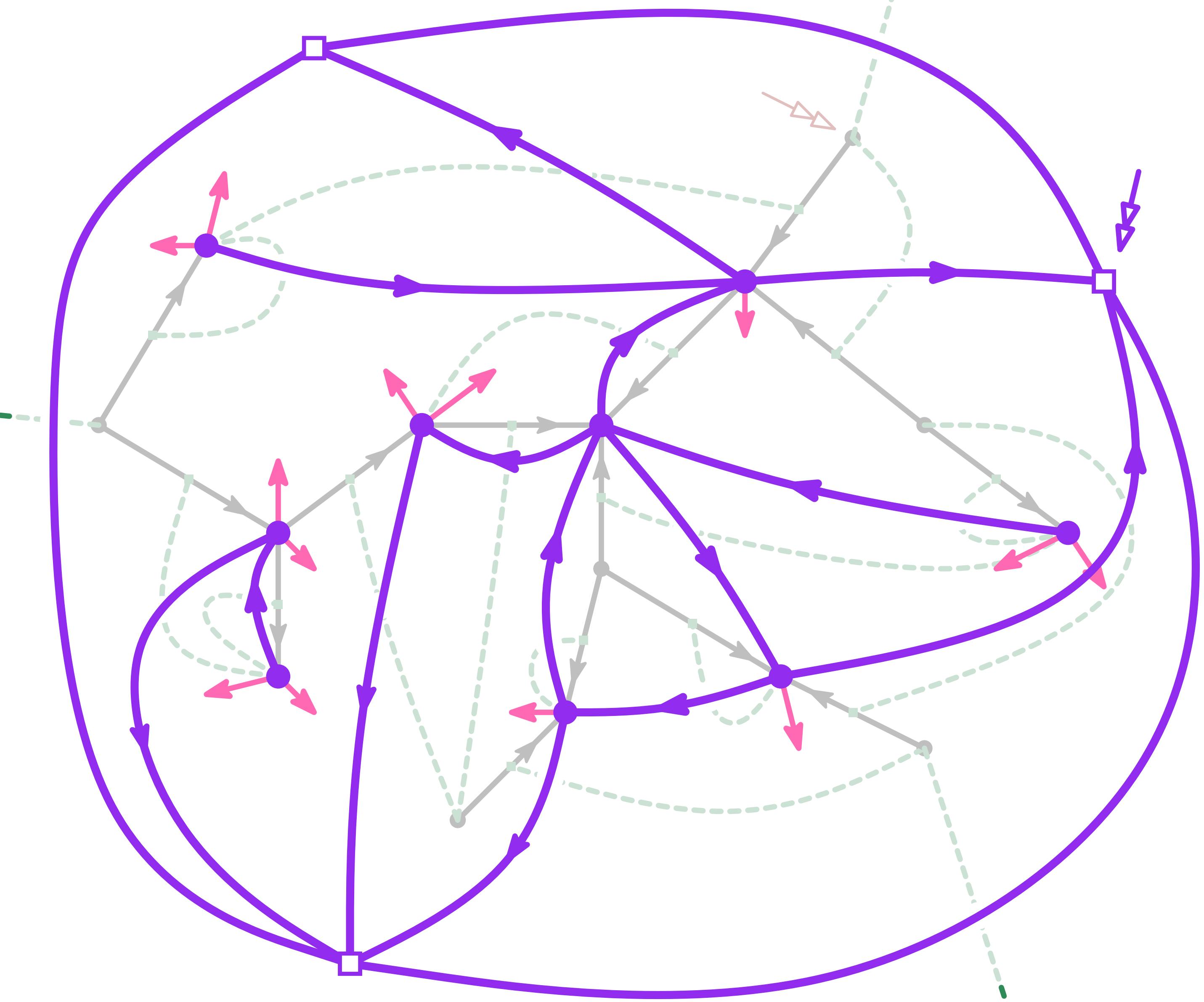


- Turning clockwise around the tree, do the following closures:

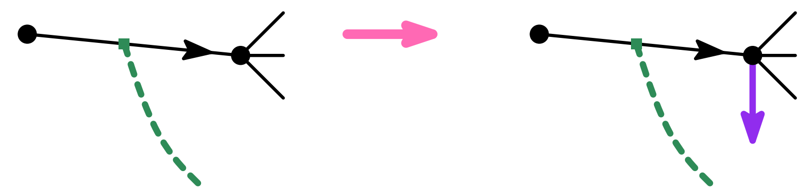


Theorem : [ABCF] This is a bijection between **outer-triangular simple maps** and **bipartite cubic maps**

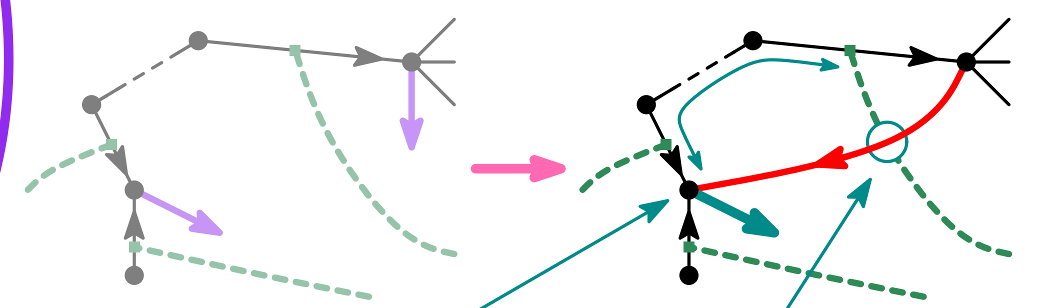
From bipartite cubic maps to simple maps



- Apply the following local rule :



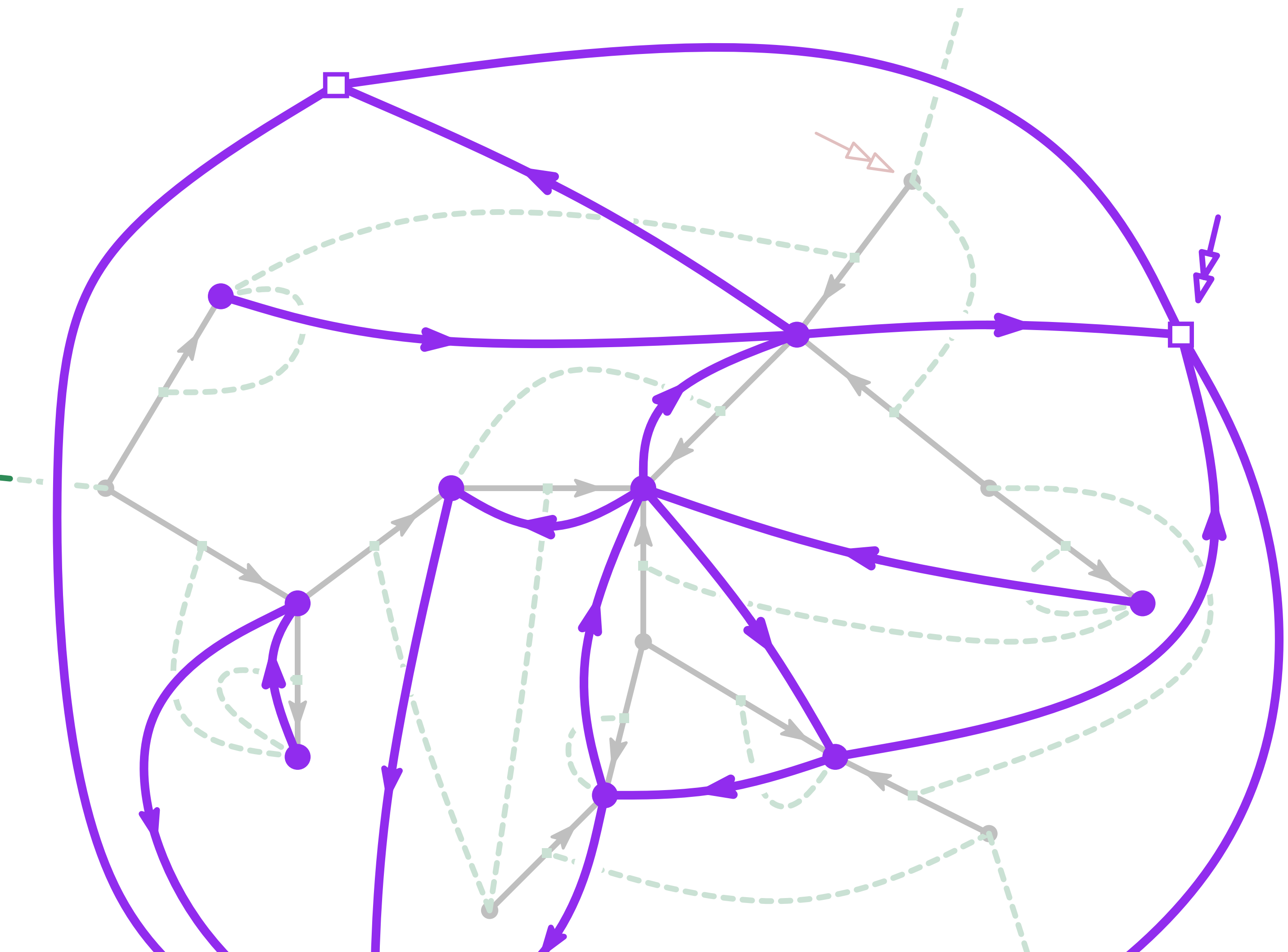
- Turning clockwise around the tree, do the following closures:



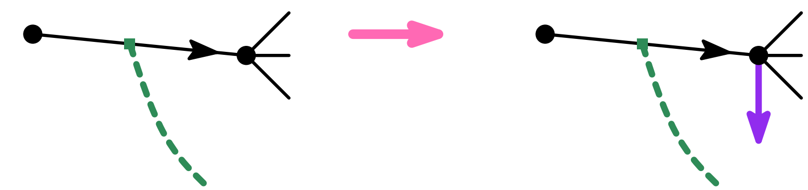
- Add 3 vertices and close the remaining opening stems sector by sector

Theorem : [ABCF] This is a bijection between **outer-triangular simple maps** and **bipartite cubic maps**

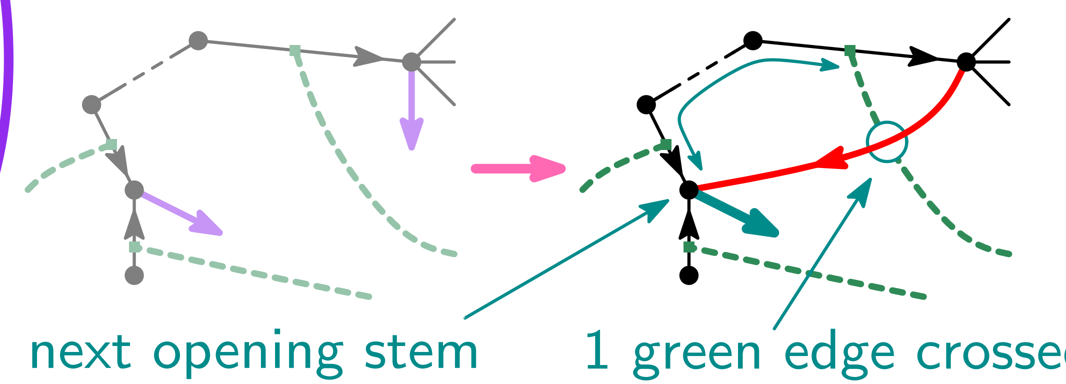
From bipartite cubic maps to simple maps



- Apply the following local rule :



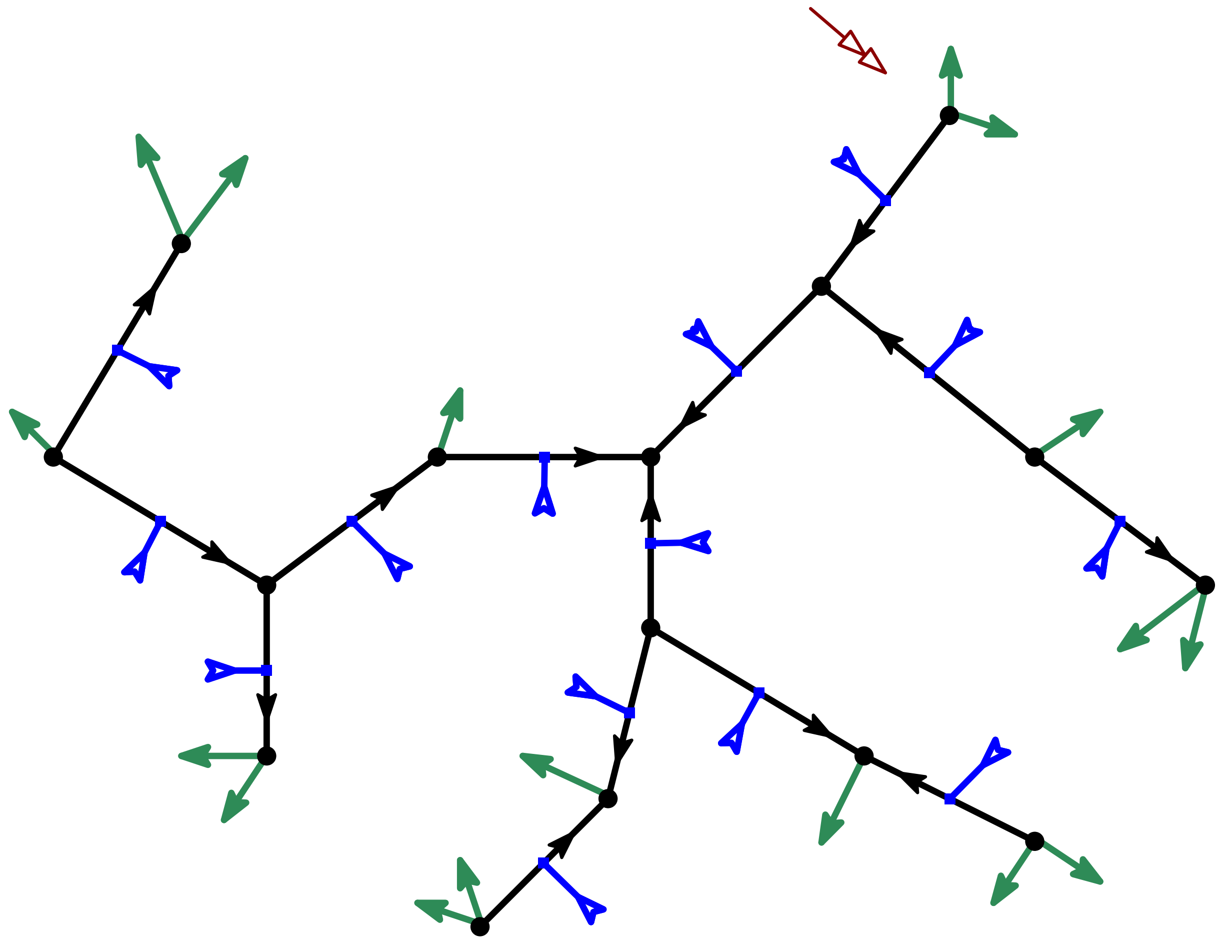
- Turning clockwise around the tree, do the following closures:



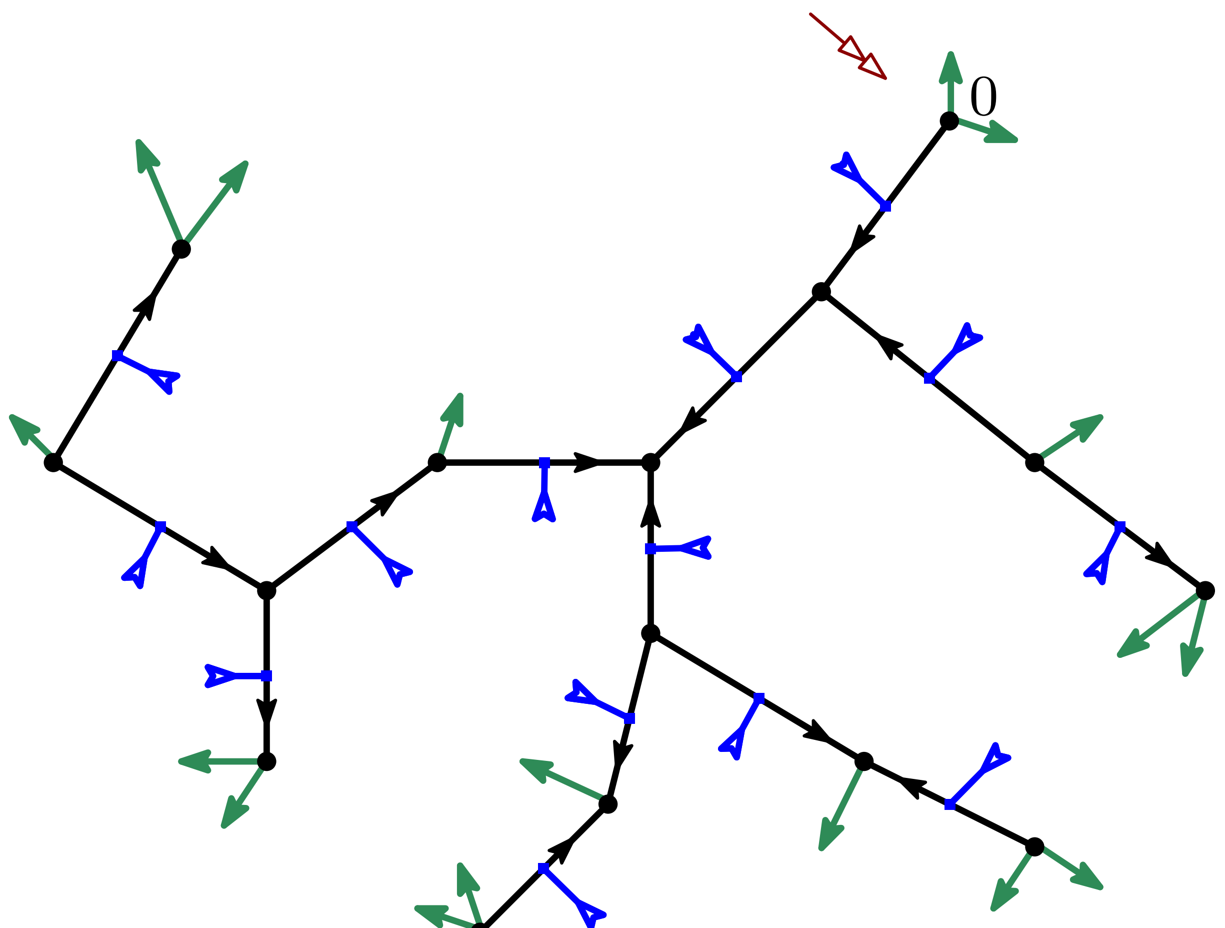
Corollary : [ABCF] We get a bijection between

outer-triangular simple maps and **balanced oriented binary trees**
(with $n+3$ edges) (with n edges)

Same bijection with labels

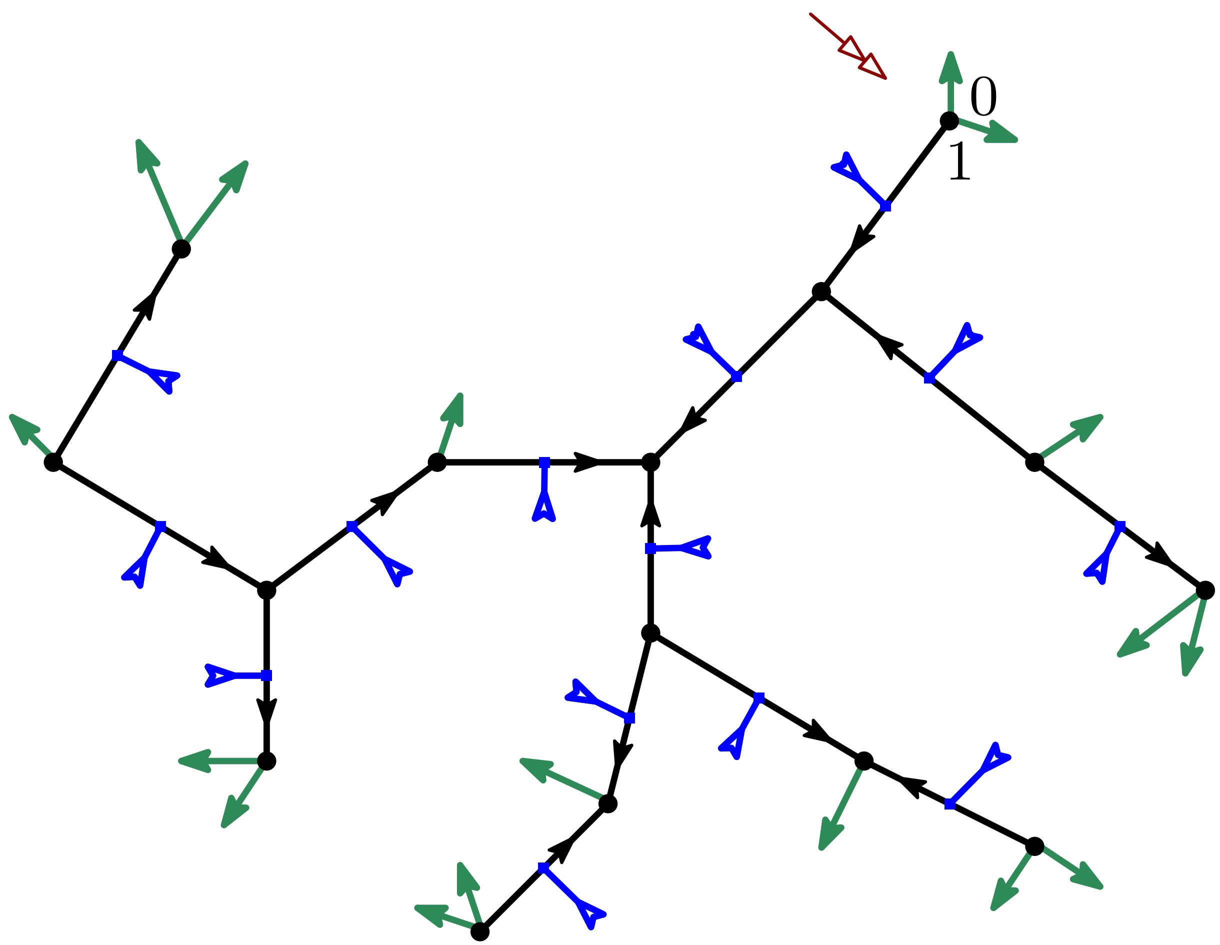


Same bijection with labels



- Label 0 the first corner.

Same bijection with labels

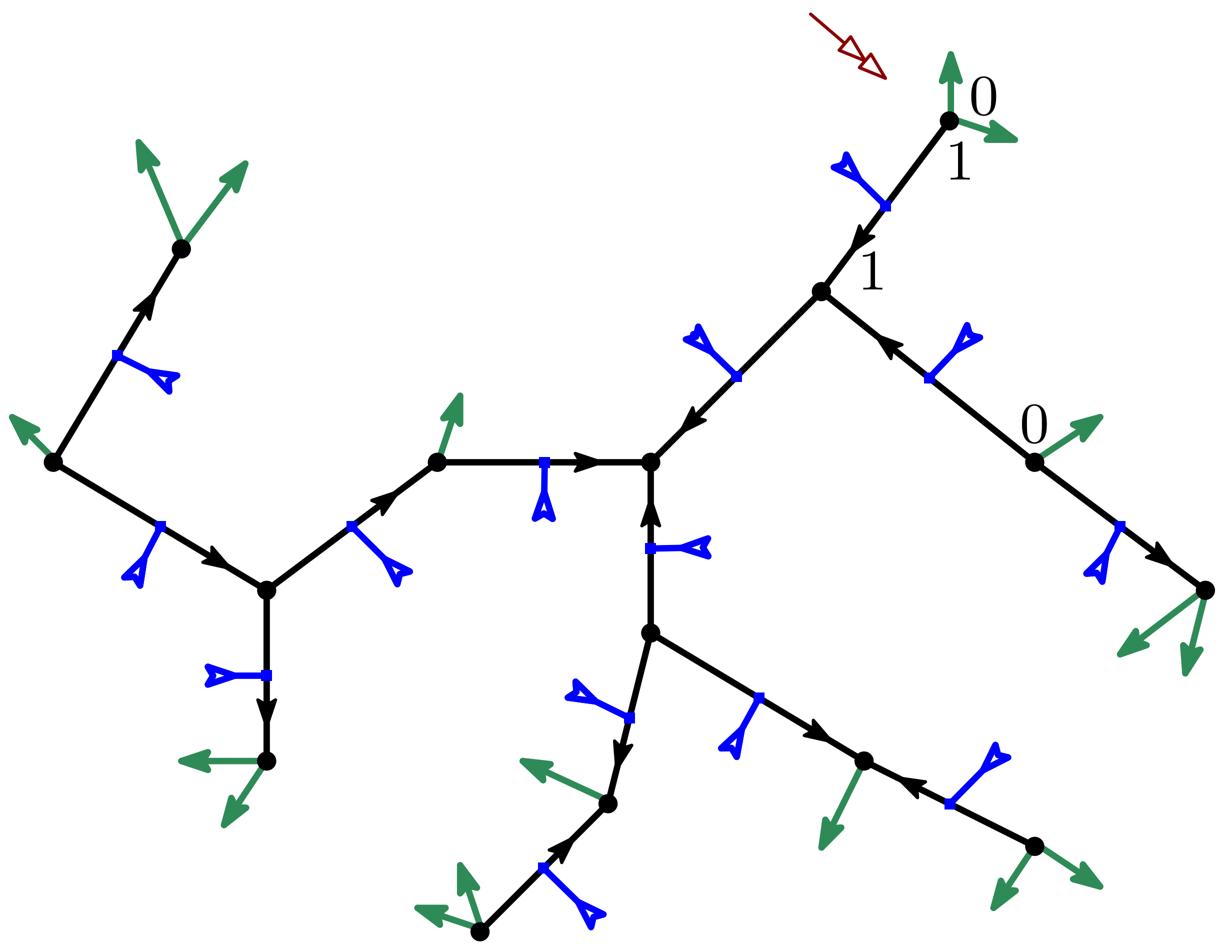


- Label 0 the first corner.

In clockwise order, apply the following rules:

- After an opening stem : increase by 1.

Same bijection with labels

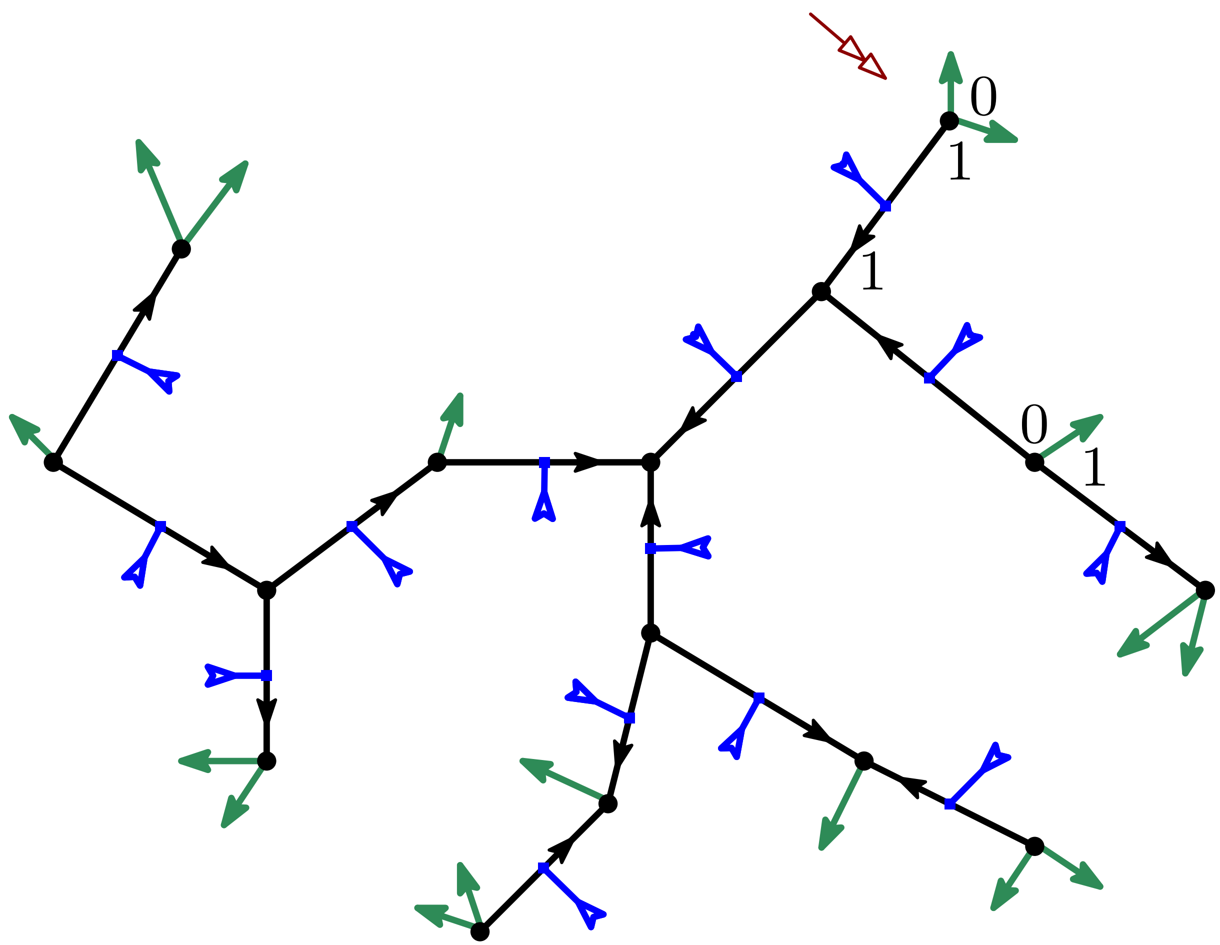


- Label 0 the first corner.

In clockwise order, apply the following rules:

- After an opening stem : increase by 1.
- After a closing stem : decrease by 1.

Same bijection with labels

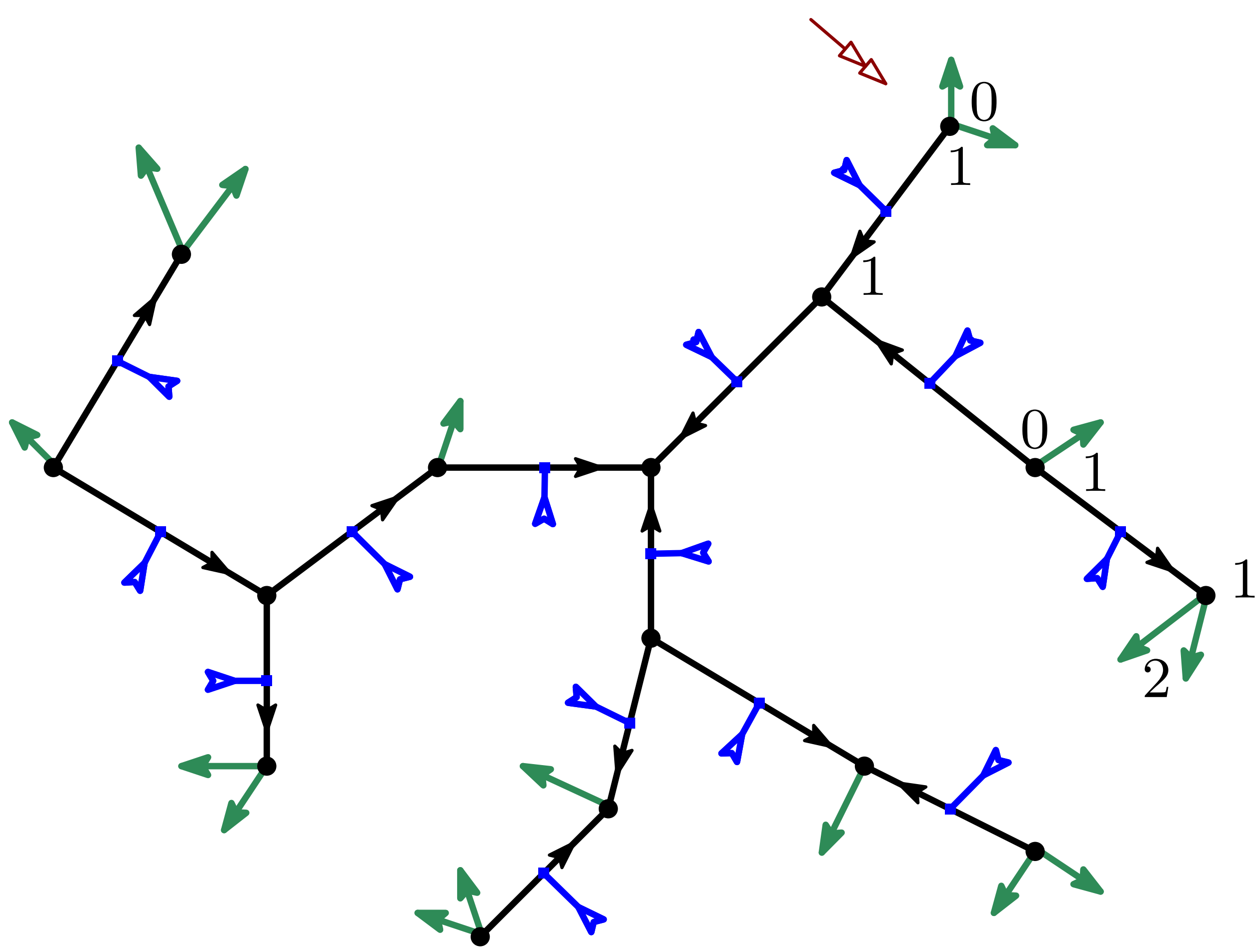


- Label 0 the first corner.

In clockwise order, apply the following rules:

- After an opening stem : increase by 1.
- After a closing stem : decrease by 1.

Same bijection with labels

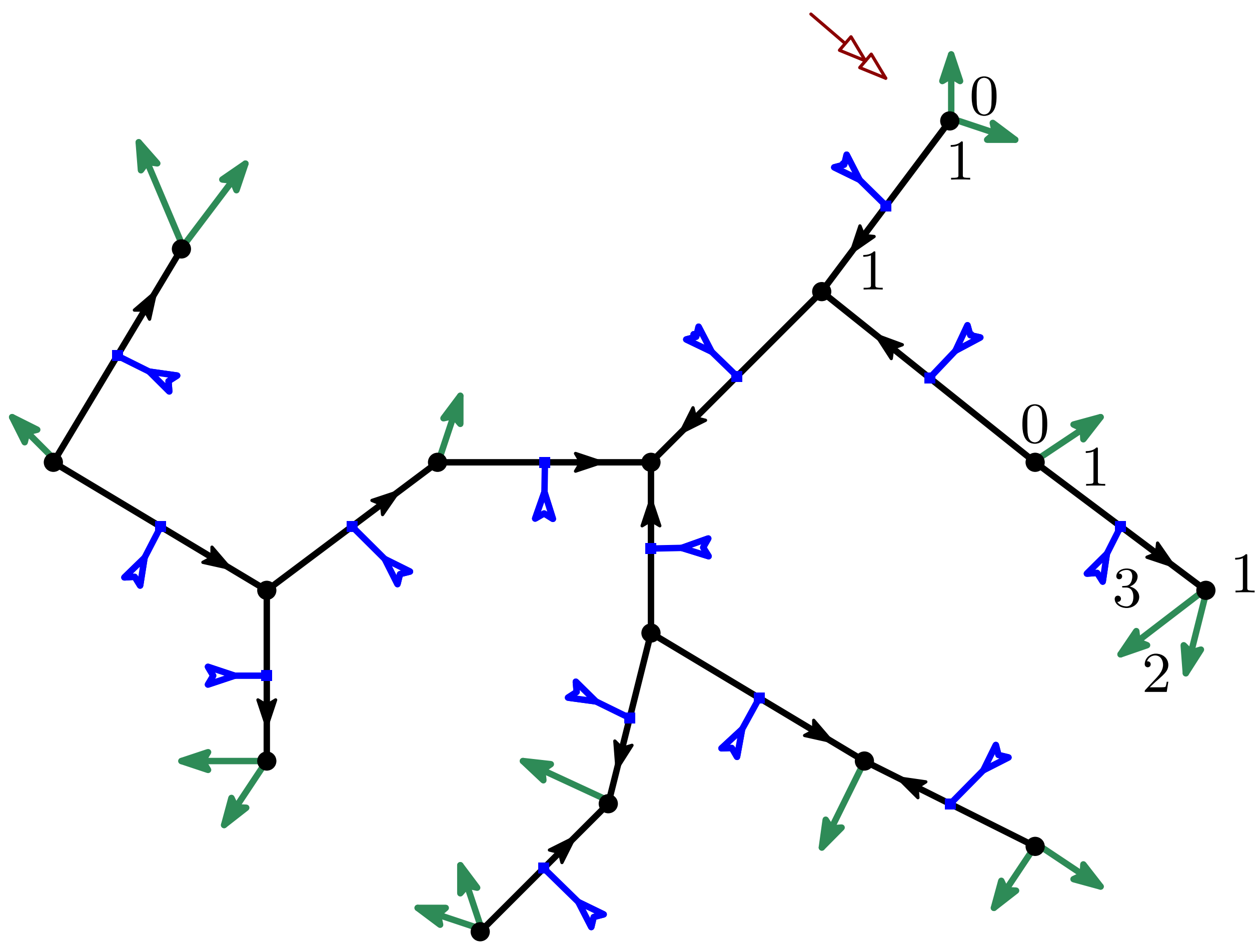


- Label 0 the first corner.

In clockwise order, apply the following rules:

- After an opening stem : increase by 1.
- After a closing stem : decrease by 1.

Same bijection with labels

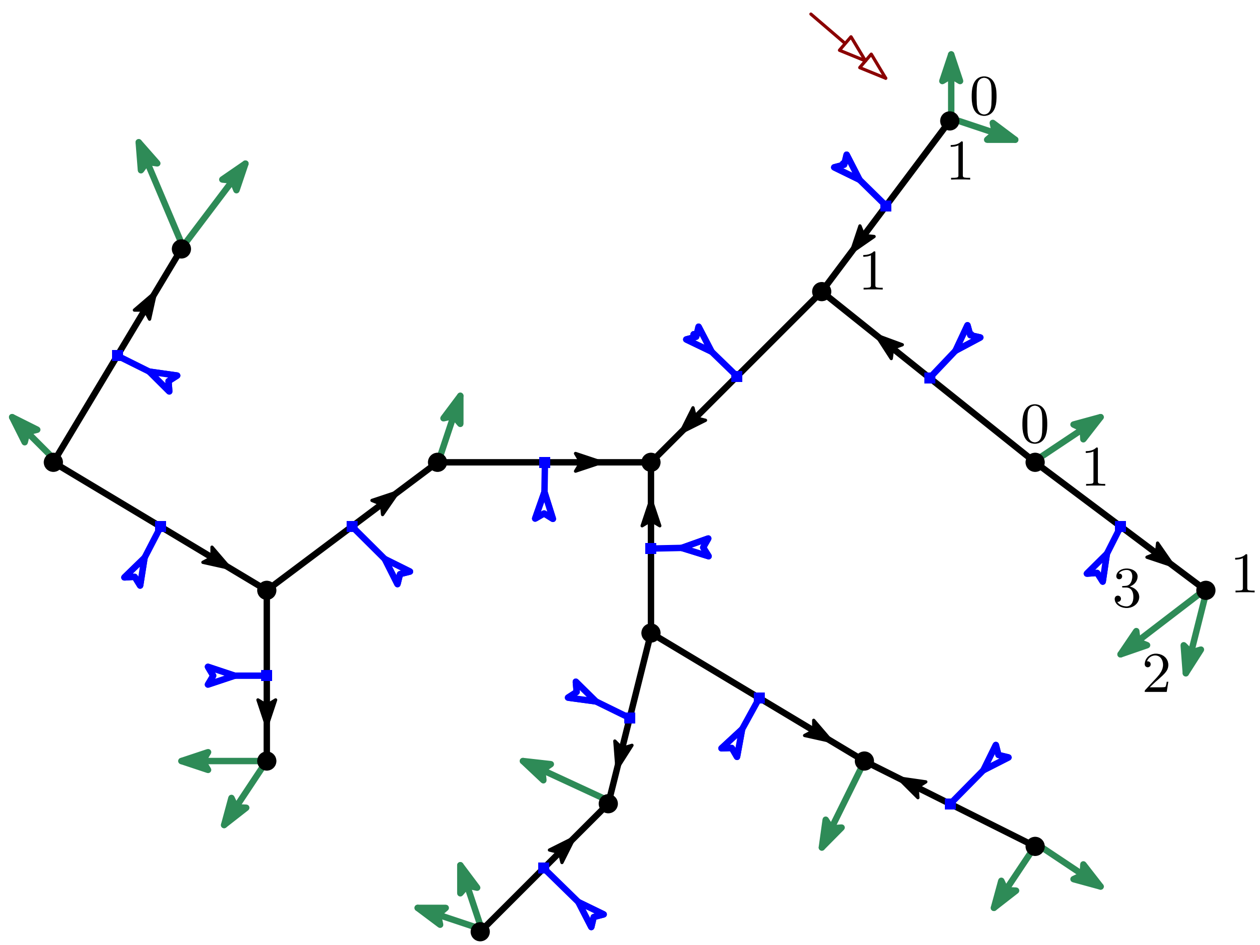


- Label 0 the first corner.

In clockwise order, apply the following rules:

- After an opening stem : increase by 1.
- After a closing stem : decrease by 1.

Same bijection with labels

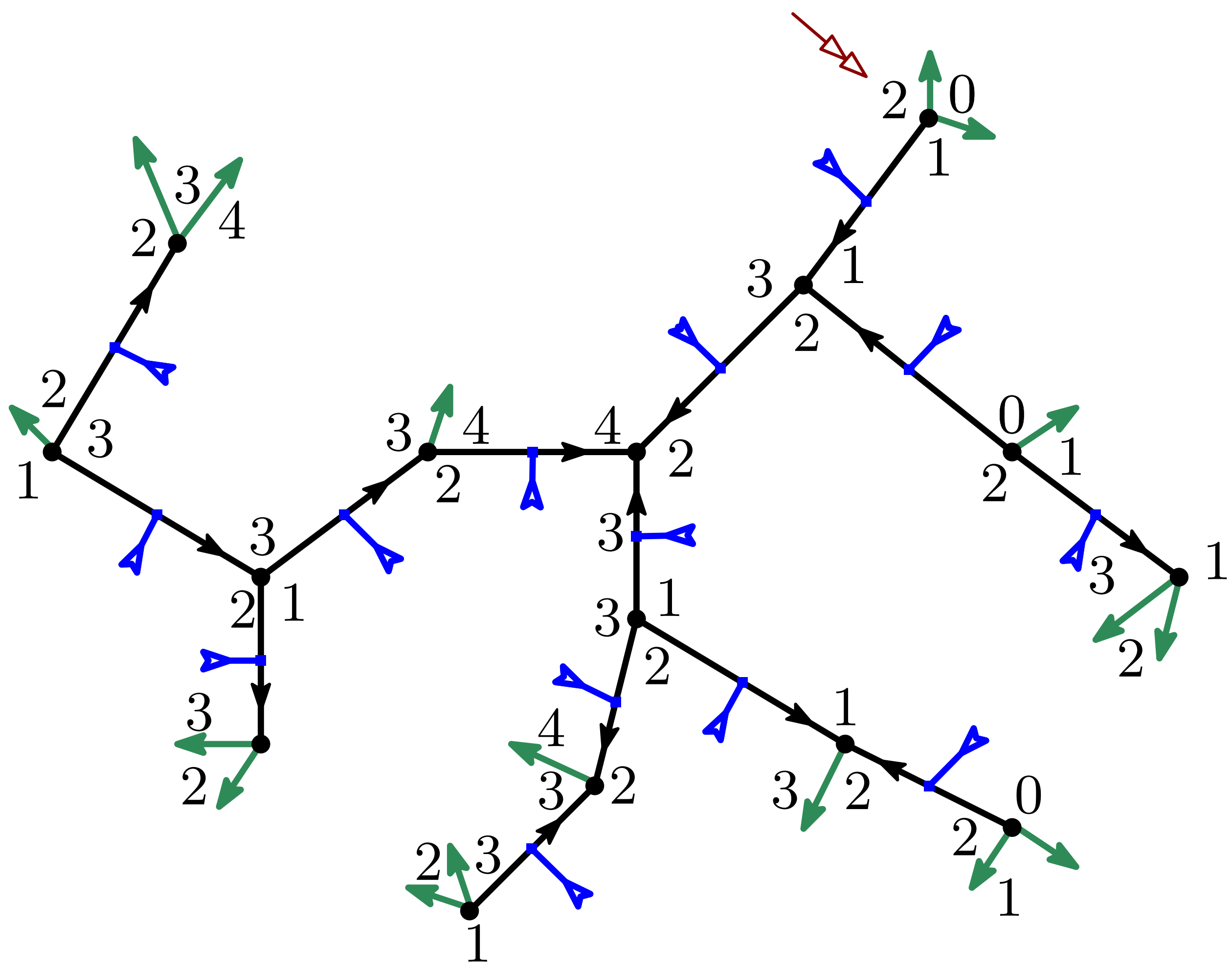


- Label 0 the first corner.

In clockwise order, apply the following rules:

- After an opening stem : increase by 1.
- After a closing stem : decrease by 1.

Same bijection with labels

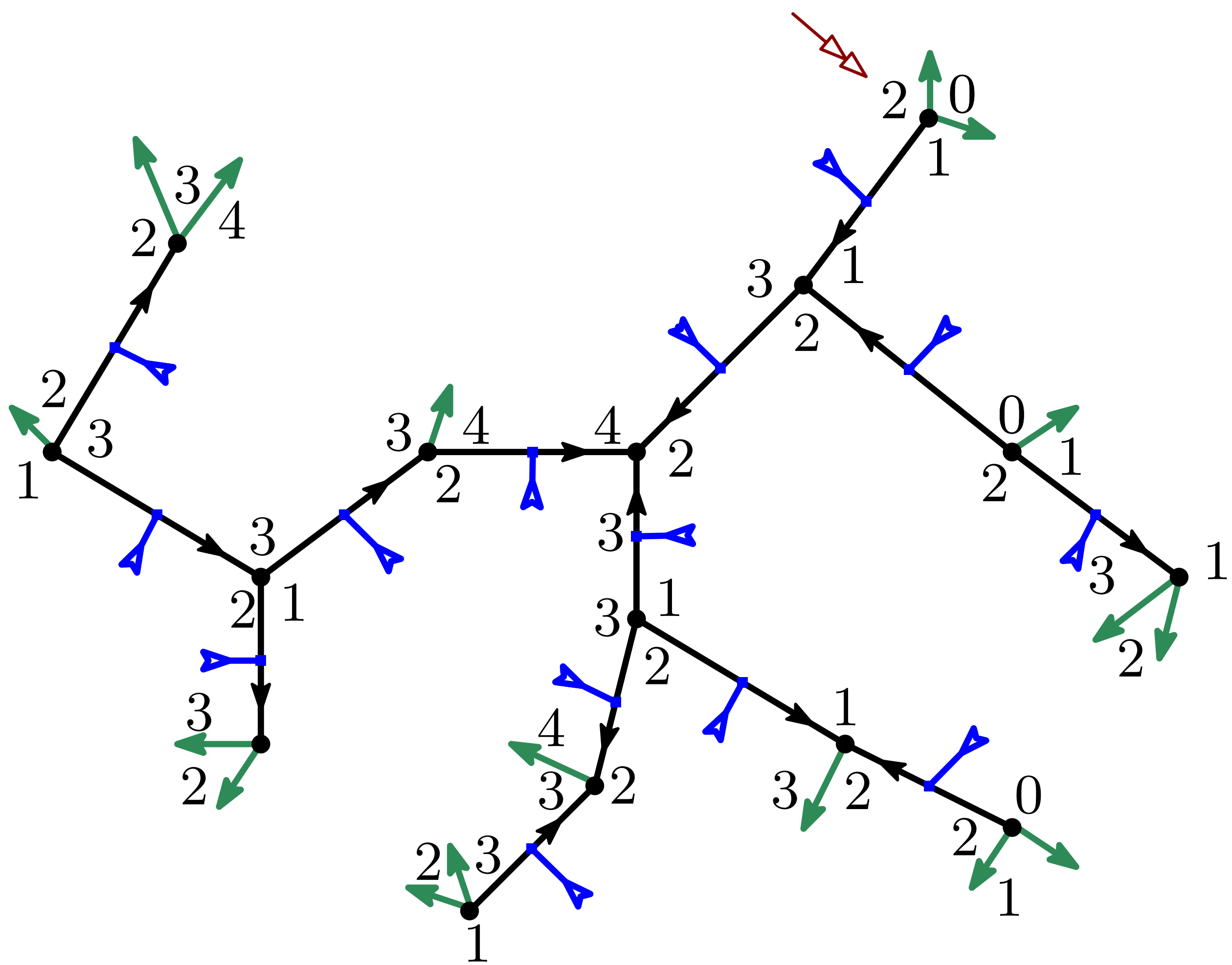


- Label 0 the first corner.

In clockwise order, apply the following rules:

- After an opening stem : increase by 1.
- After a closing stem : decrease by 1.

Same bijection with labels



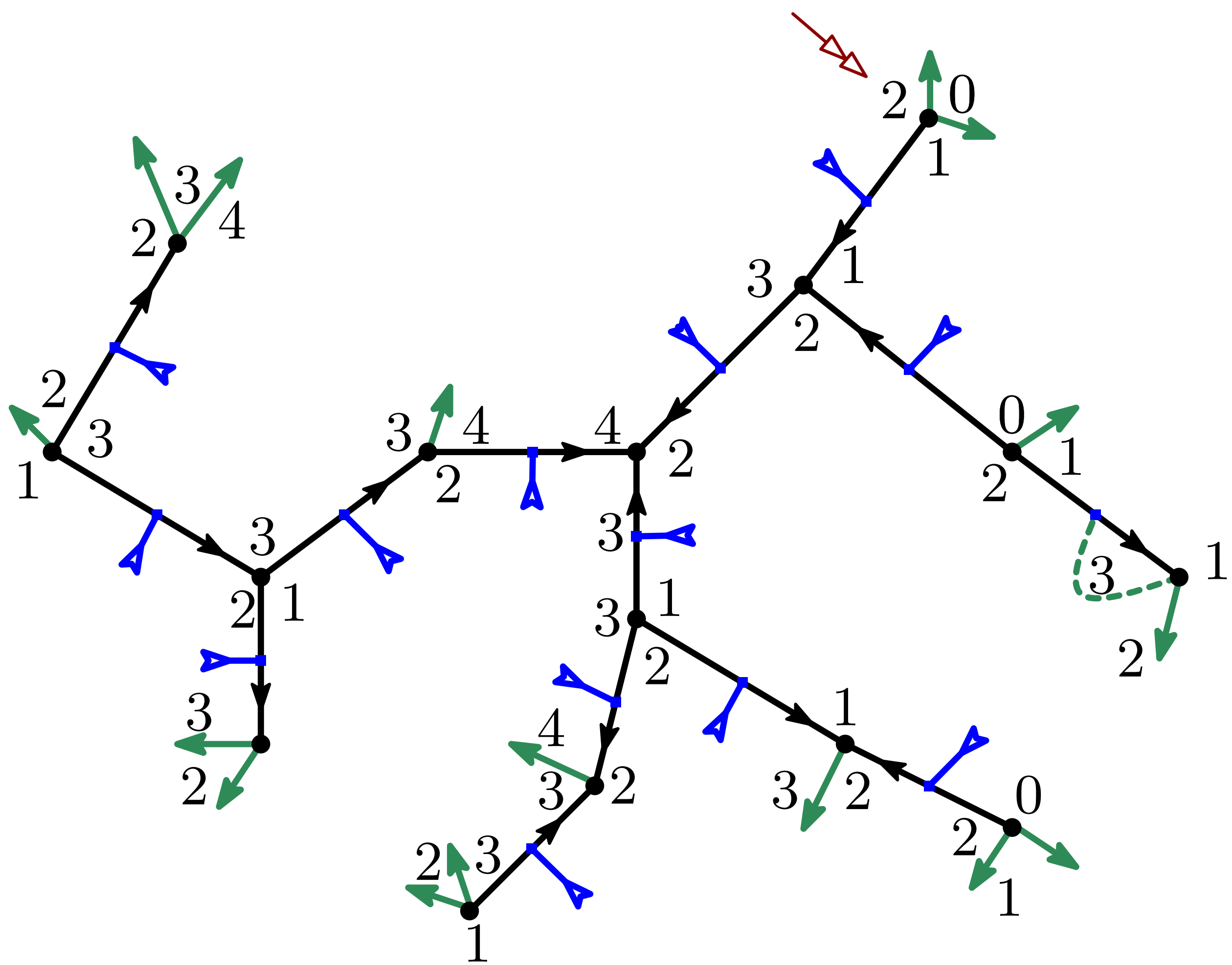
- Label 0 the first corner.

In clockwise order, apply the following rules:

- After an opening stem : increase by 1.
- After a closing stem : decrease by 1.

Labels $\approx$ depth of the face in the cubic map

Same bijection with labels



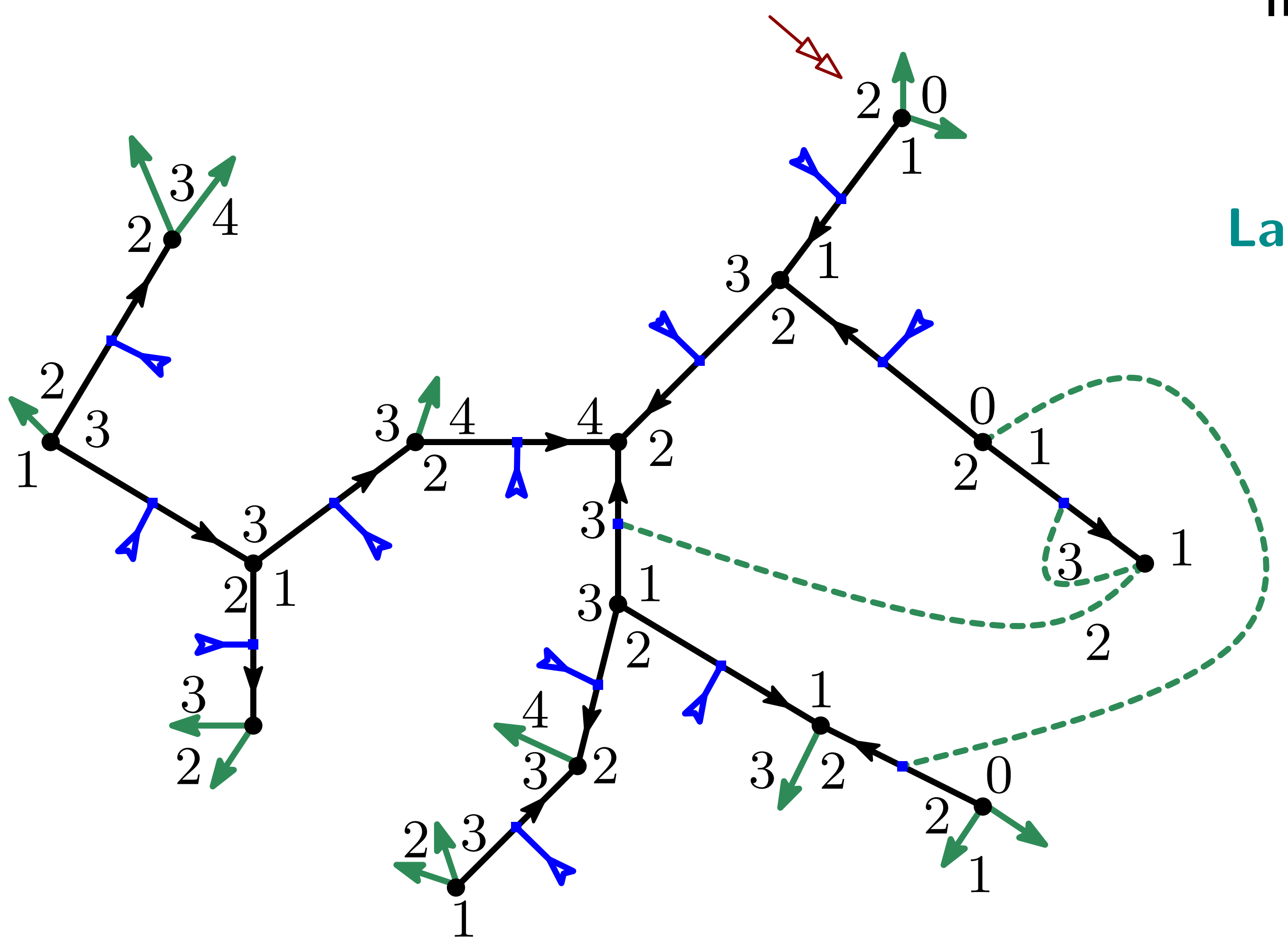
- Label 0 the first corner.

In clockwise order, apply the following rules:

- After an opening stem : increase by 1.
- After a closing stem : decrease by 1.

Labels $\approx$ depth of the face in the cubic map

Same bijection with labels



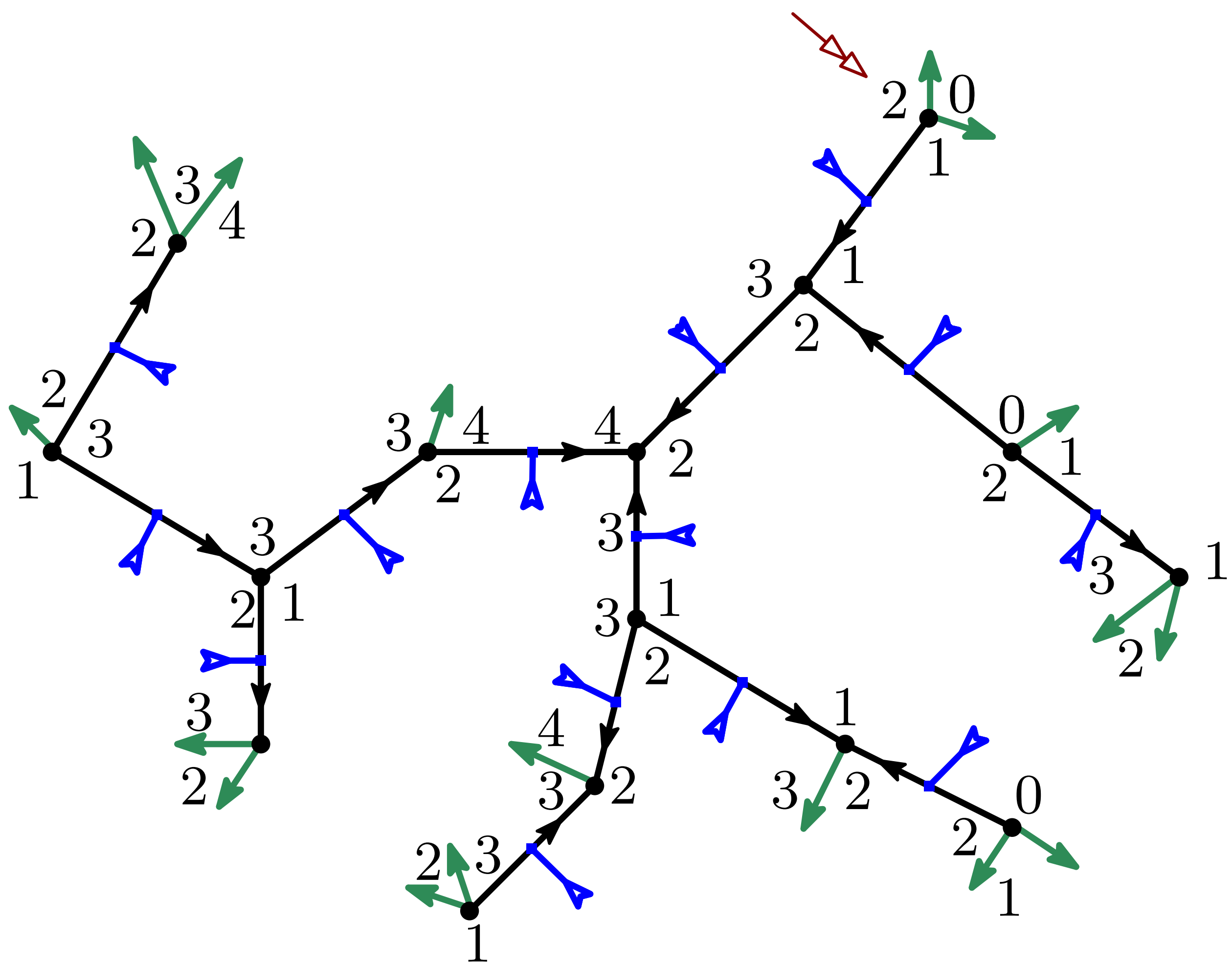
- Label 0 the first corner.

In clockwise order, apply the following rules:

- After an opening stem : increase by 1.
- After a closing stem : decrease by 1.

Labels ≈ depth of the face in the cubic map

Same bijection with labels



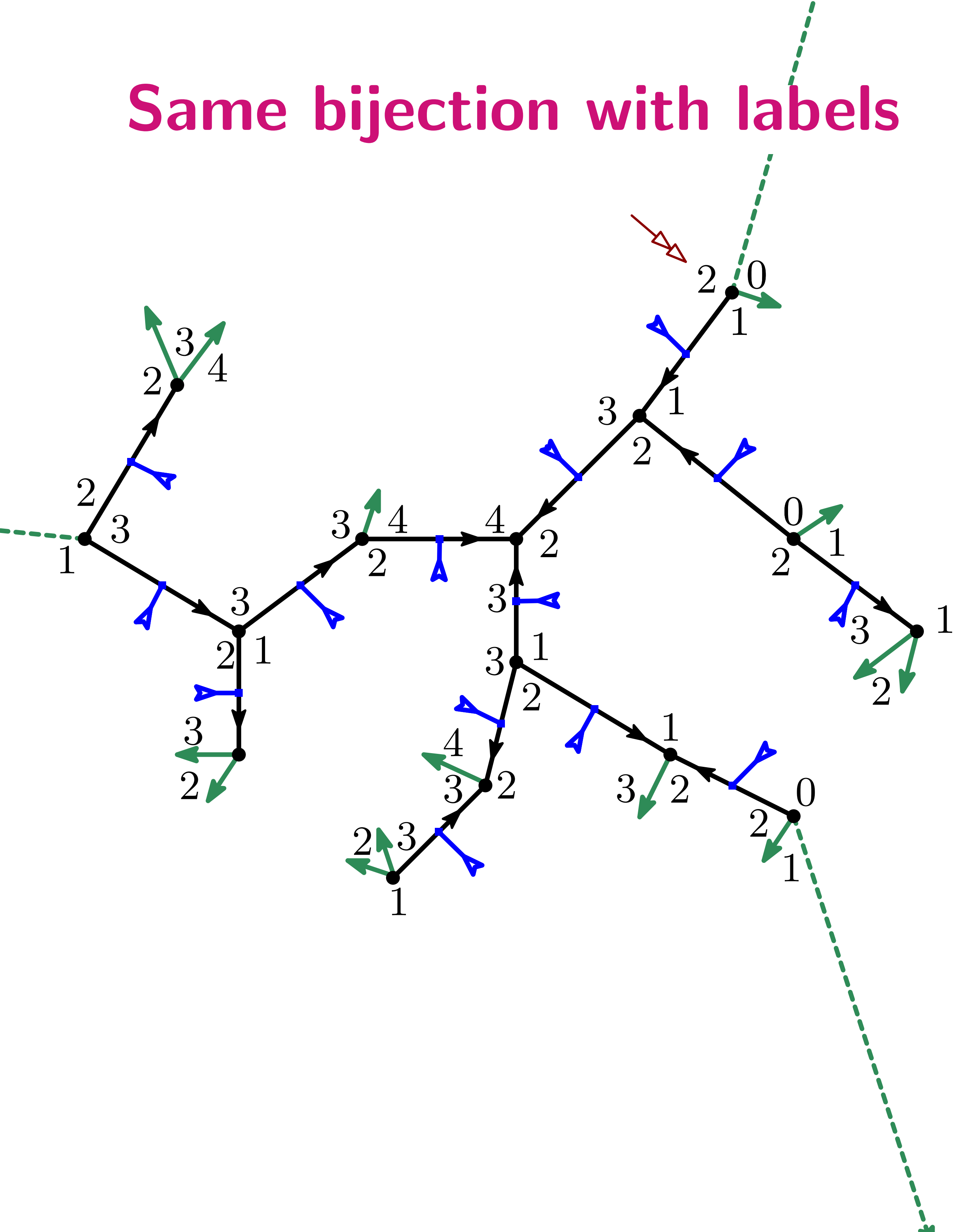
- Label 0 the first corner.

In clockwise order, apply the following rules:

- After an opening stem : increase by 1.
- After a closing stem : decrease by 1.

Labels $\approx$ depth of the face in the cubic map

Same bijection with labels



- Label 0 the first corner.

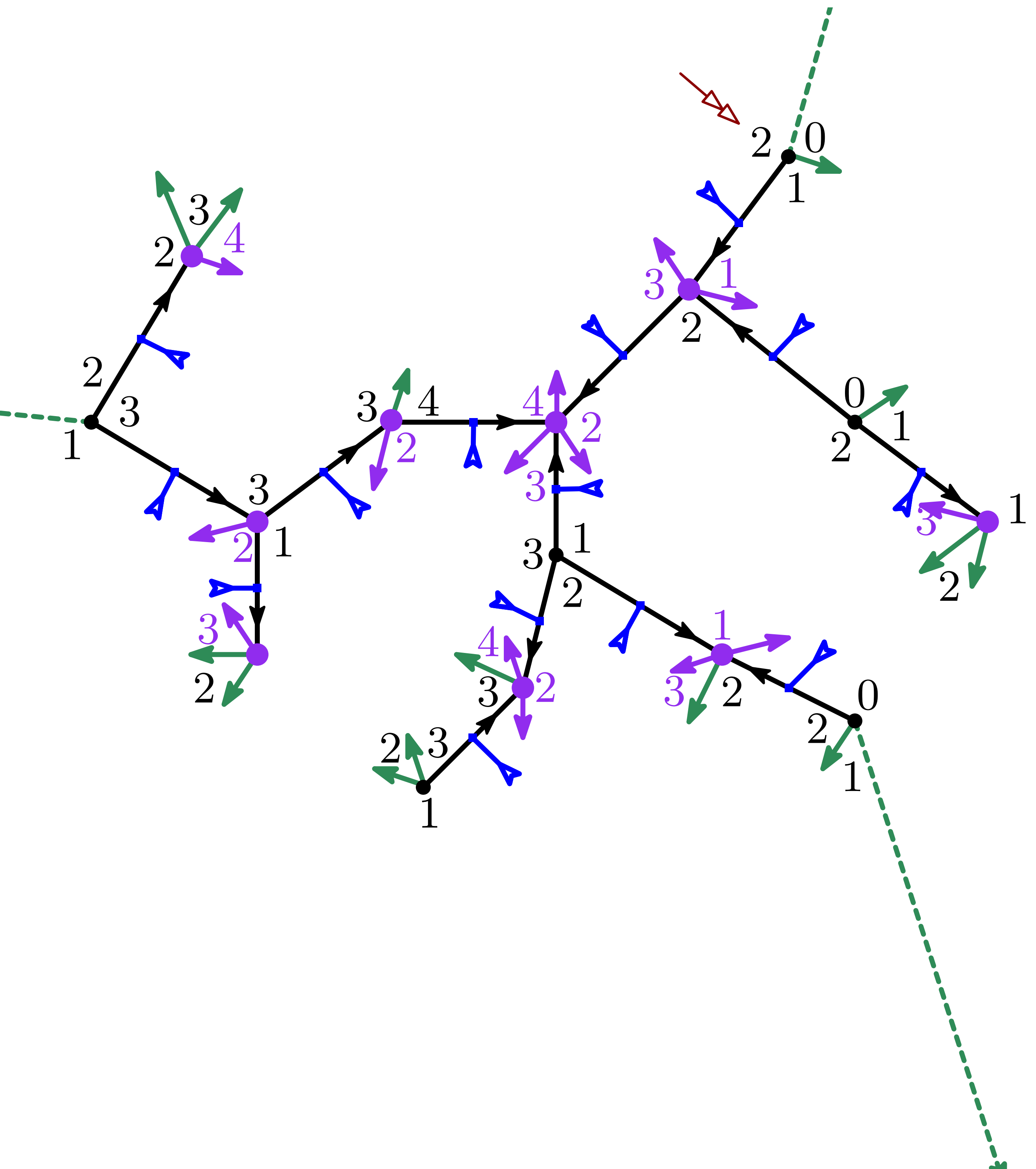
In clockwise order, apply the following rules:

- After an opening stem : increase by 1.
- After a closing stem : decrease by 1.

Labels $\approx$ depth of the face in the cubic map

Unmatched stems = last 0, 1 and 2 corners

Same bijection with labels



- Label 0 the first corner.

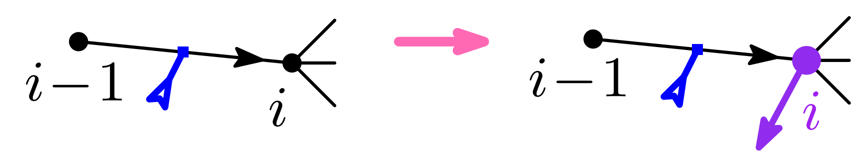
In clockwise order, apply the following rules:

- After an opening stem : increase by 1.
- After a closing stem : decrease by 1.

Labels ≈ depth of the face in the cubic map

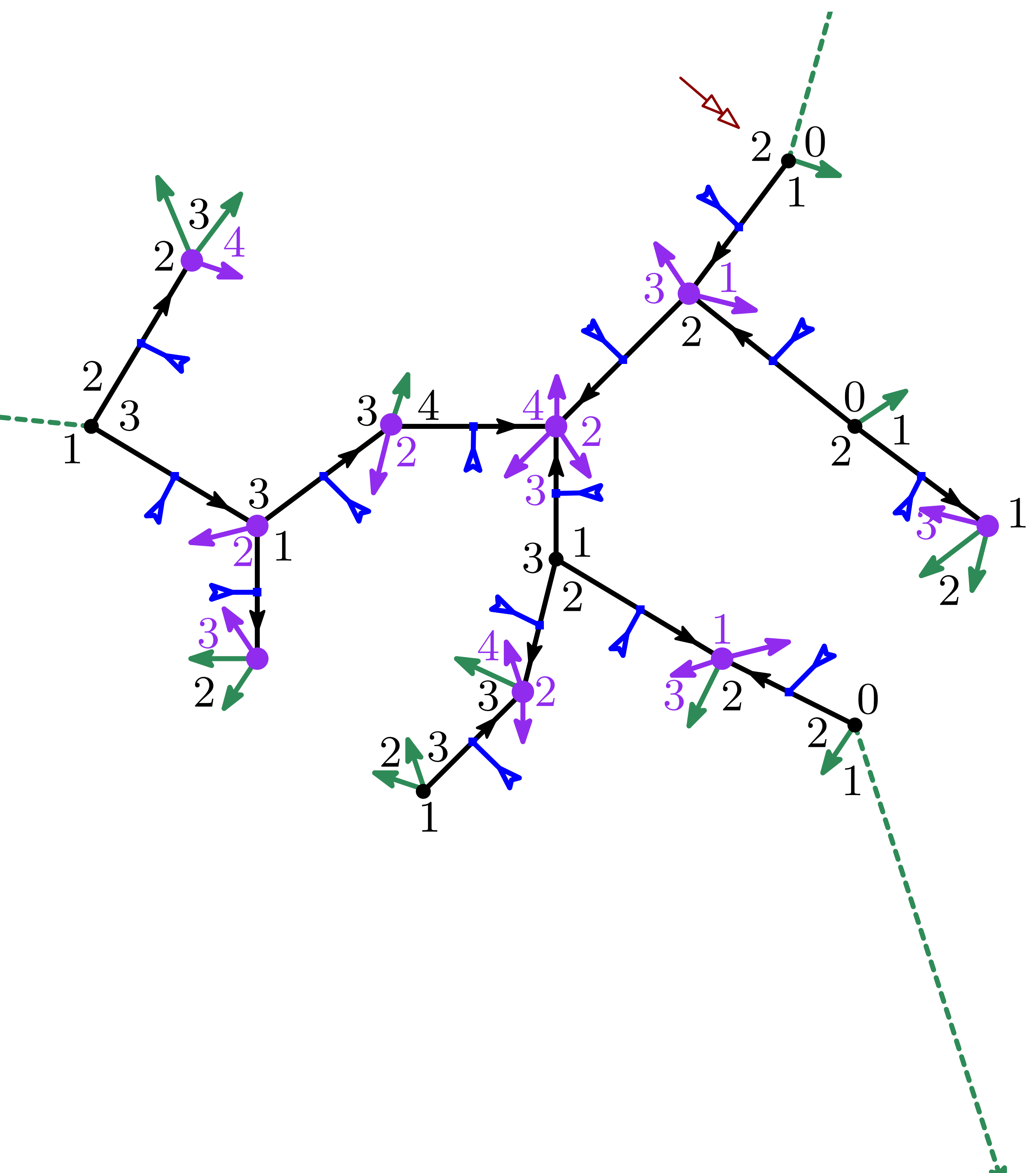
Unmatched stems = last 0, 1 and 2 corners

- Apply the following local rule :



(= add a → before each descent and color the corresponding corner and vertex)

Same bijection with labels



- Label 0 the first corner.

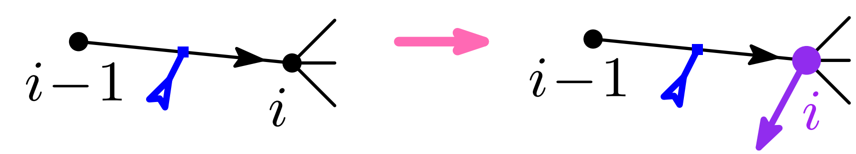
In clockwise order, apply the following rules:

- After an opening stem : increase by 1.
- After a closing stem : decrease by 1.

Labels ≈ depth of the face in the cubic map

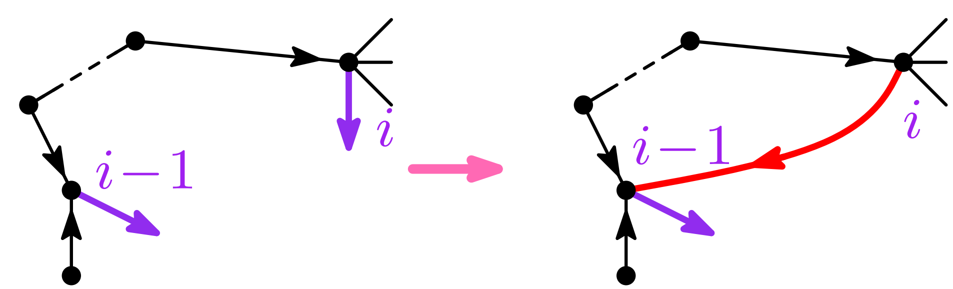
Unmatched stems = last 0, 1 and 2 corners

- Apply the following local rule :

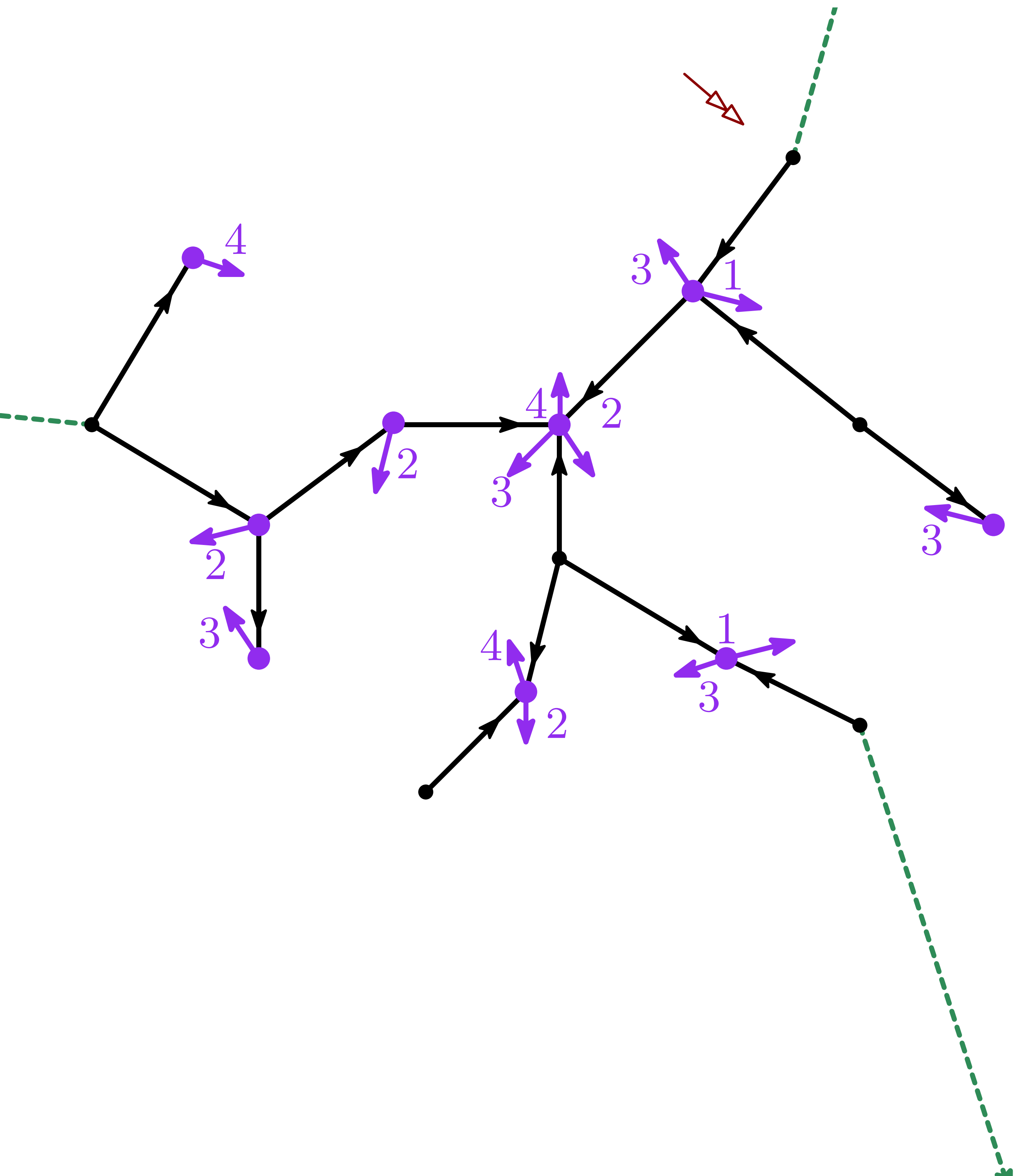


(= add a → before each descent and color the corresponding corner and vertex)

- Erase all non-purple and do the following closures



Same bijection with labels



- Label 0 the first corner.

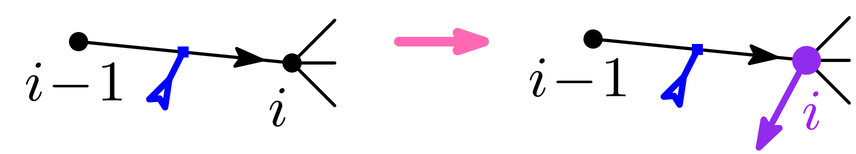
In clockwise order, apply the following rules:

- After an opening stem : increase by 1.
- After a closing stem : decrease by 1.

Labels ≈ depth of the face in the cubic map

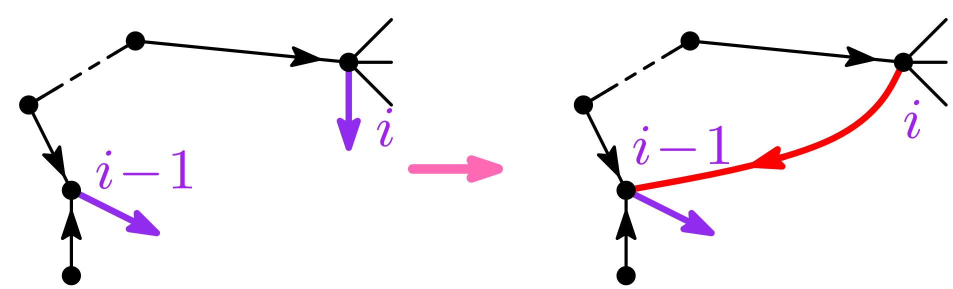
Unmatched stems = last 0, 1 and 2 corners

- Apply the following local rule :

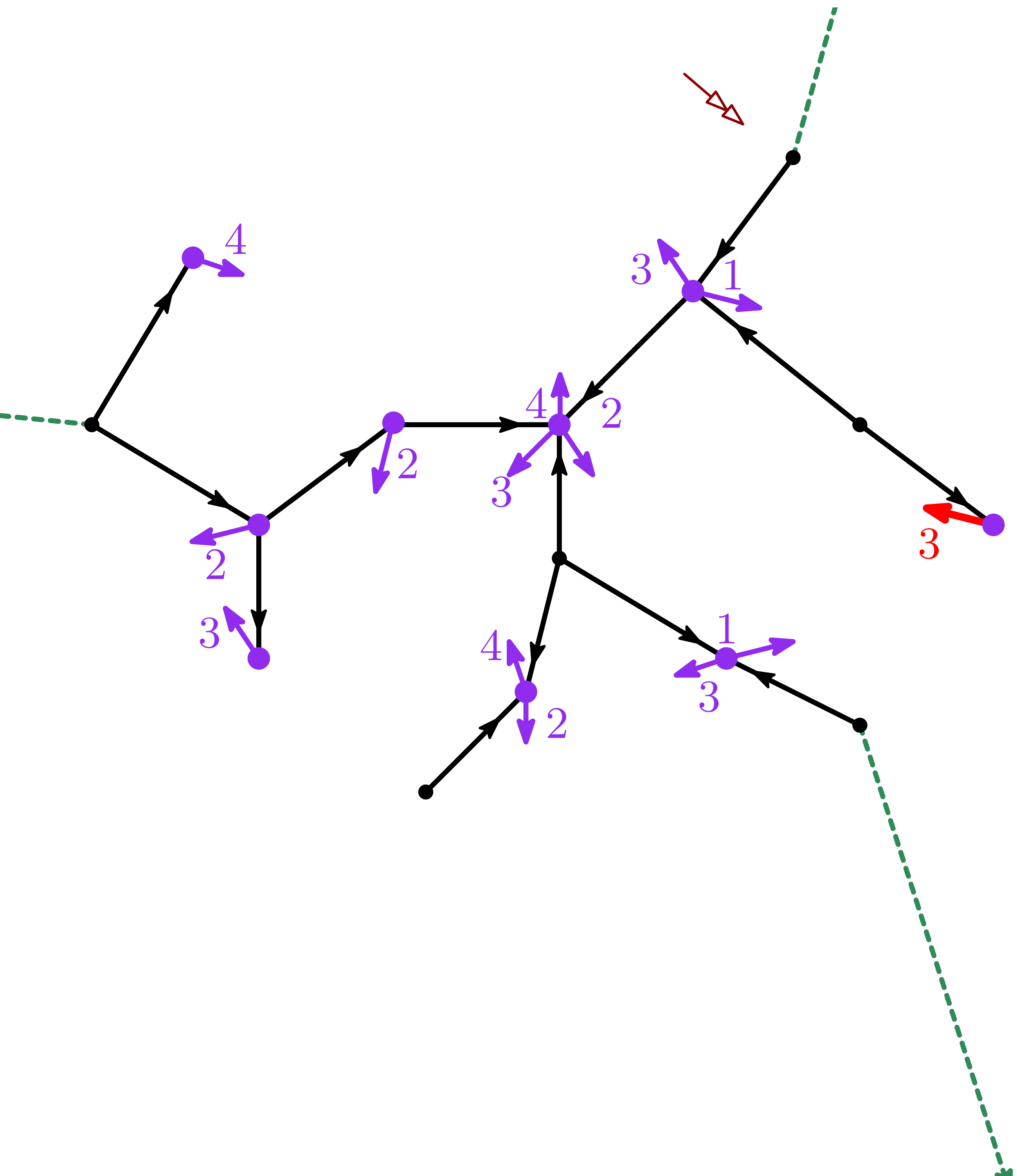


(= add a → before each descent and color the corresponding corner and vertex)

- Erase all non-purple and do the following closures



Same bijection with labels



- Label 0 the first corner.

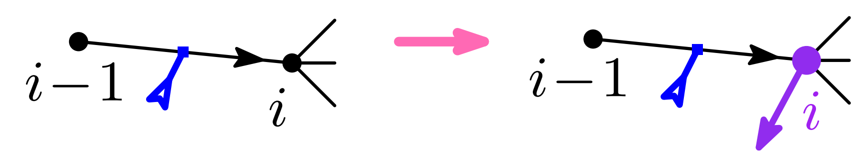
In clockwise order, apply the following rules:

- After an opening stem : increase by 1.
- After a closing stem : decrease by 1.

Labels $\approx$ depth of the face in the cubic map

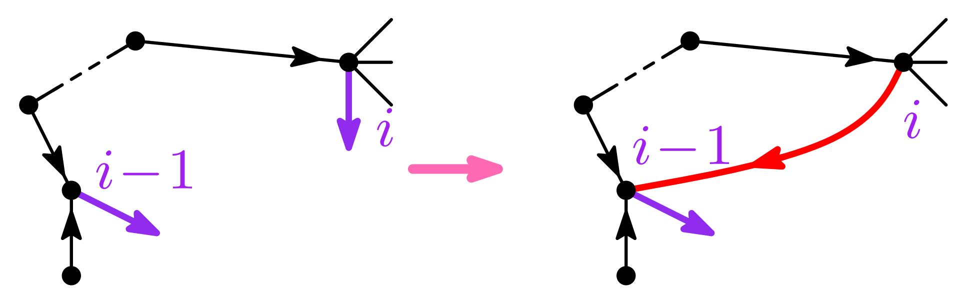
Unmatched stems = last 0, 1 and 2 corners

- Apply the following local rule :

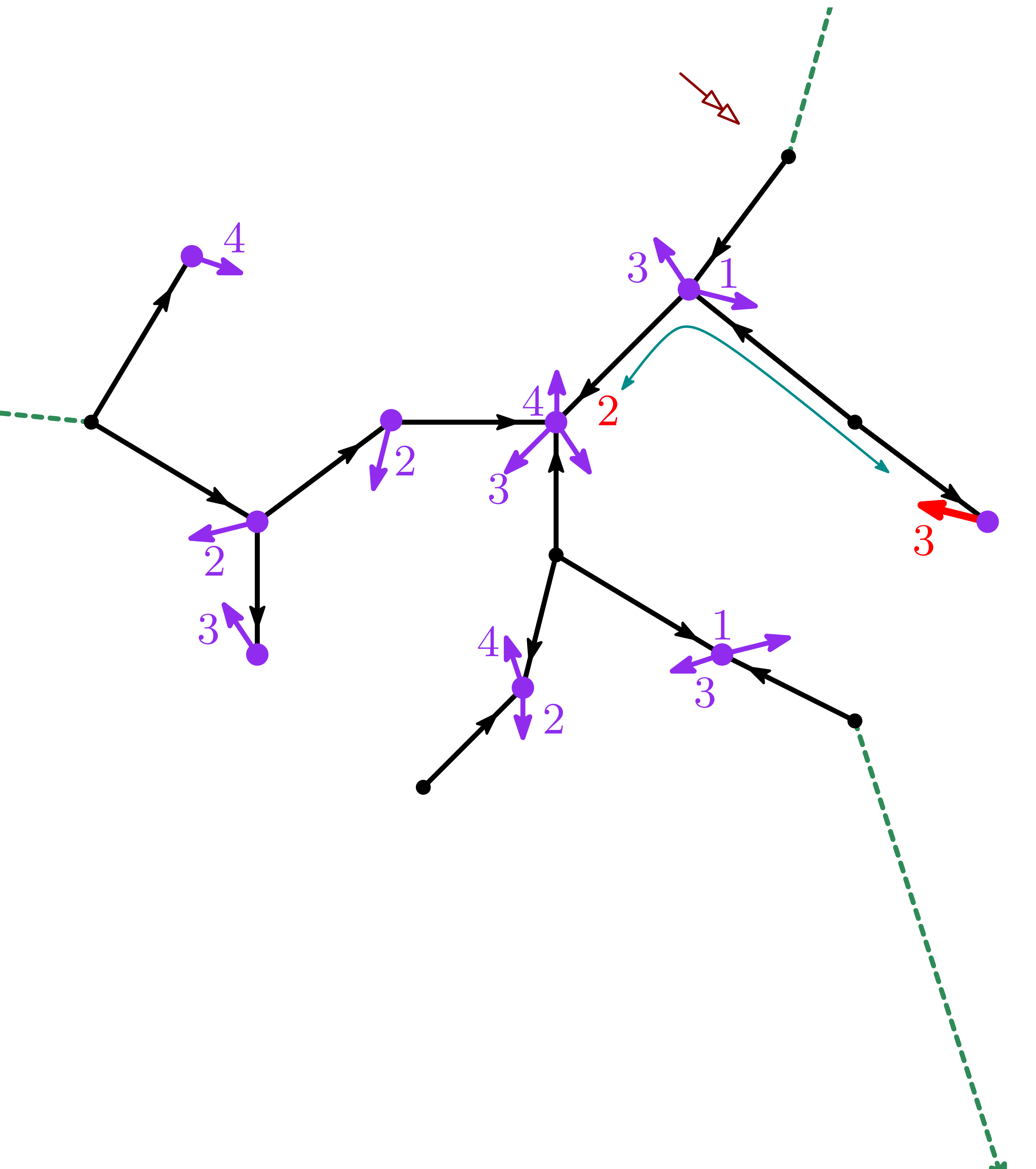


(= add a before each descent and color the corresponding corner and vertex)

- Erase all non-purple and do the following closures



Same bijection with labels



- Label 0 the first corner.

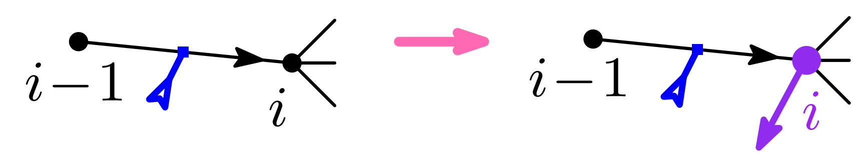
In clockwise order, apply the following rules:

- After an opening stem : increase by 1.
- After a closing stem : decrease by 1.

Labels $\approx$ depth of the face in the cubic map

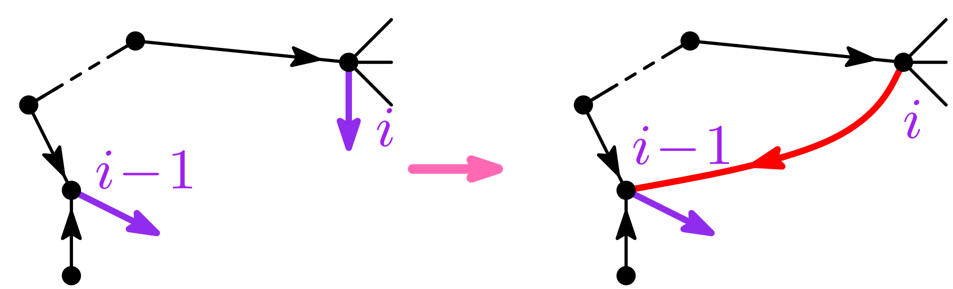
Unmatched stems = last 0, 1 and 2 corners

- Apply the following local rule :

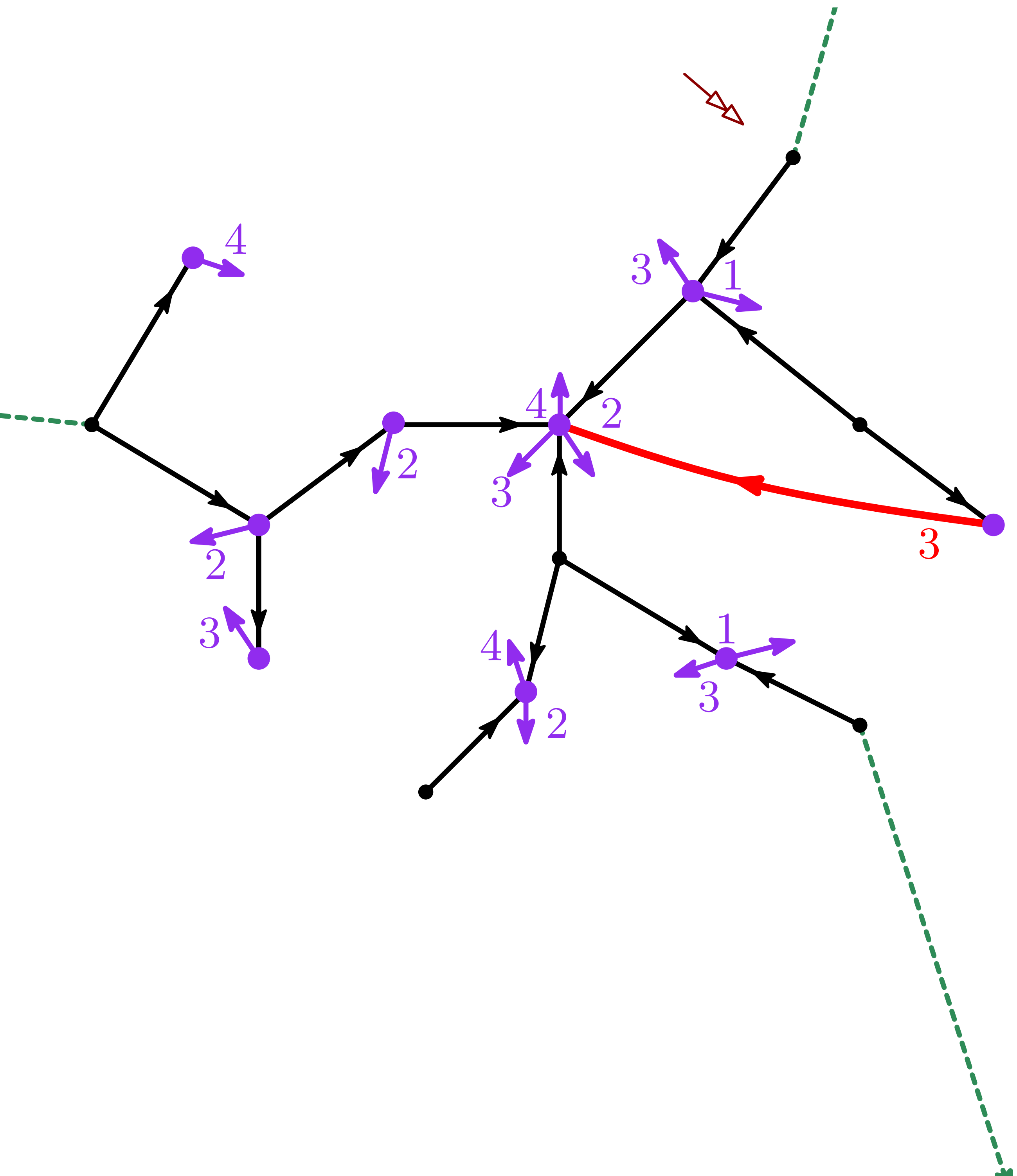


(= add a → before each descent and color the corresponding corner and vertex)

- Erase all non-purple and do the following closures



Same bijection with labels



- Label 0 the first corner.

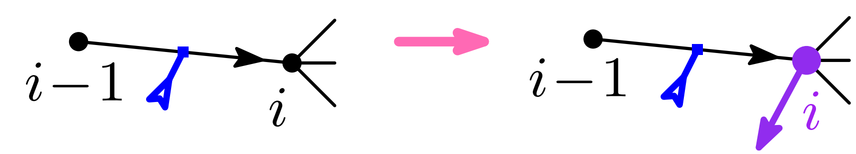
In clockwise order, apply the following rules:

- After an opening stem : increase by 1.
- After a closing stem : decrease by 1.

Labels ≈ depth of the face in the cubic map

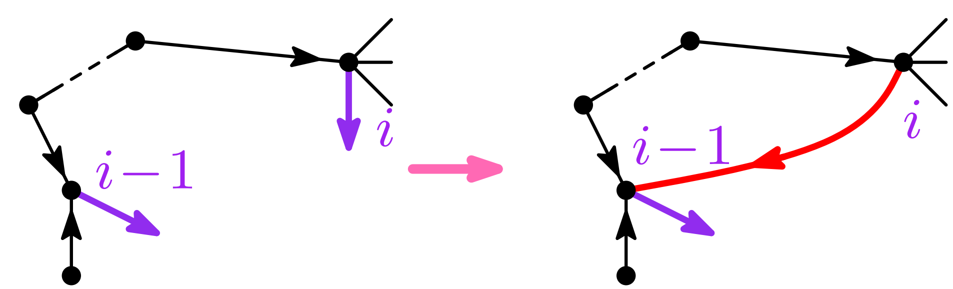
Unmatched stems = last 0, 1 and 2 corners

- Apply the following local rule :

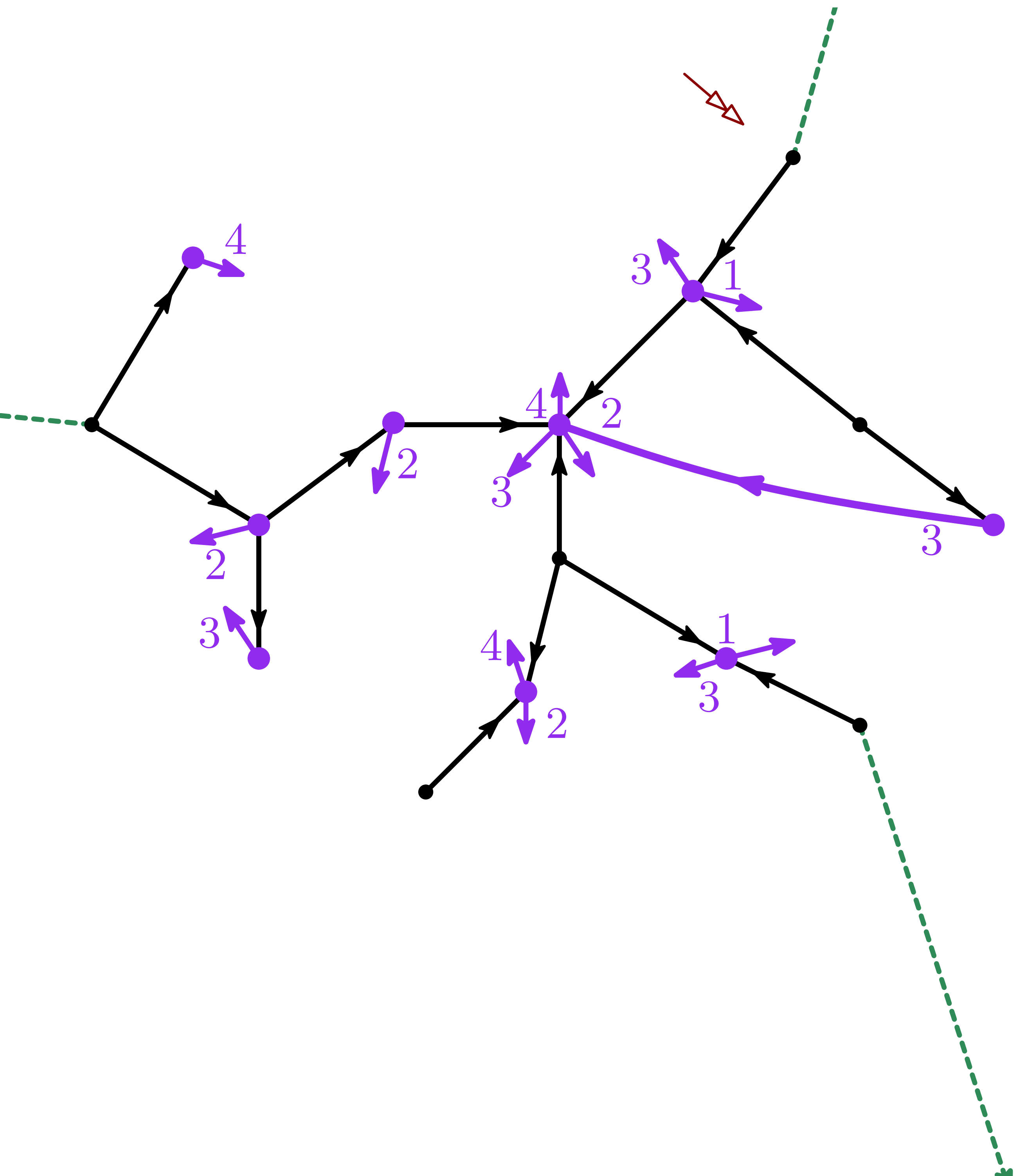


(= add a → before each descent and color the corresponding corner and vertex)

- Erase all non-purple and do the following closures



Same bijection with labels



- Label 0 the first corner.

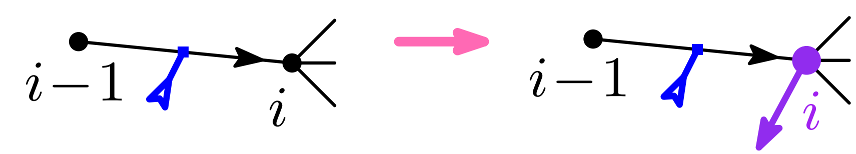
In clockwise order, apply the following rules:

- After an opening stem : increase by 1.
- After a closing stem : decrease by 1.

Labels $\approx$ depth of the face in the cubic map

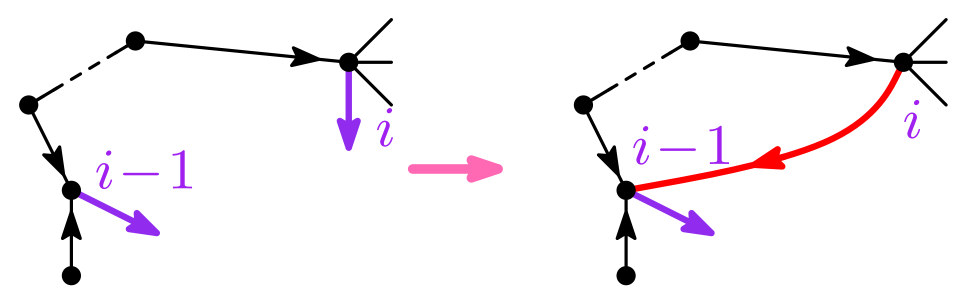
Unmatched stems = last 0, 1 and 2 corners

- Apply the following local rule :

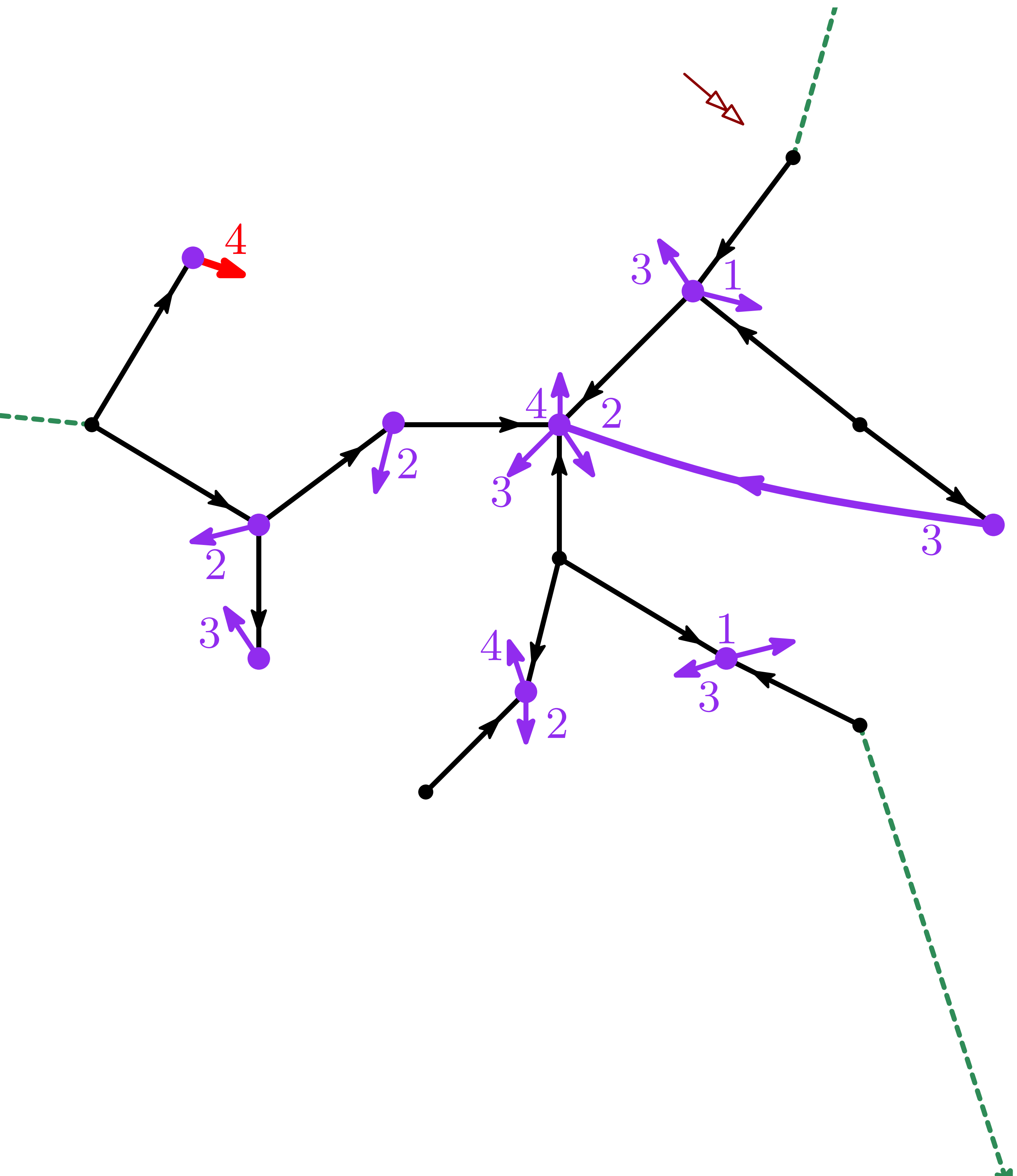


(= add a → before each descent and color the corresponding corner and vertex)

- Erase all non-purple and do the following closures



Same bijection with labels



- Label 0 the first corner.

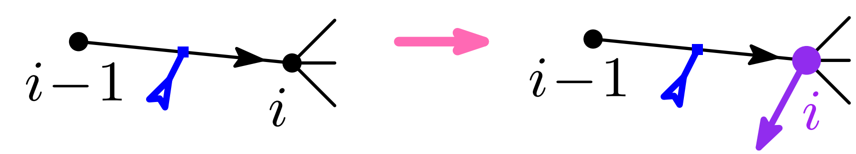
In clockwise order, apply the following rules:

- After an opening stem : increase by 1.
- After a closing stem : decrease by 1.

Labels $\approx$ depth of the face in the cubic map

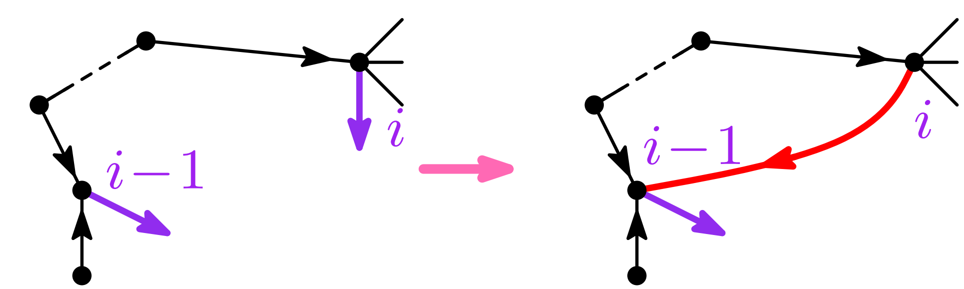
Unmatched stems = last 0, 1 and 2 corners

- Apply the following local rule :

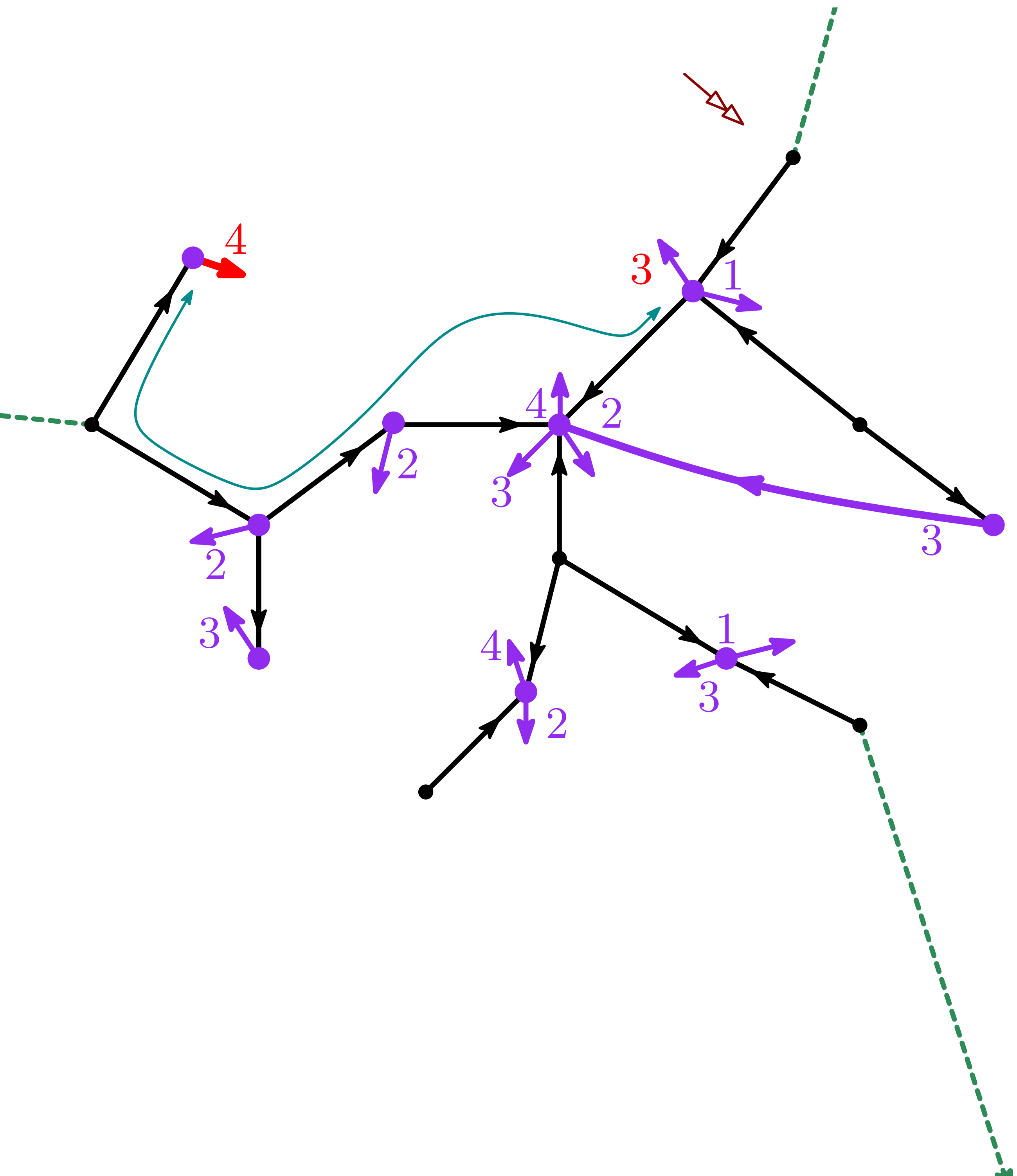


(= add a → before each descent and color the corresponding corner and vertex)

- Erase all non-purple and do the following closures



Same bijection with labels



- Label 0 the first corner.

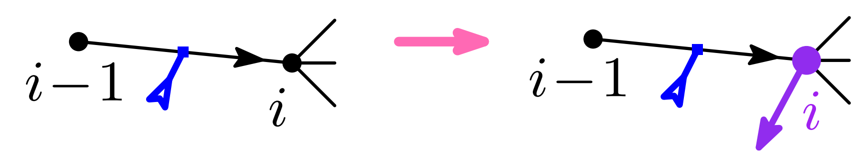
In clockwise order, apply the following rules:

- After an opening stem : increase by 1.
- After a closing stem : decrease by 1.

Labels ≈ depth of the face in the cubic map

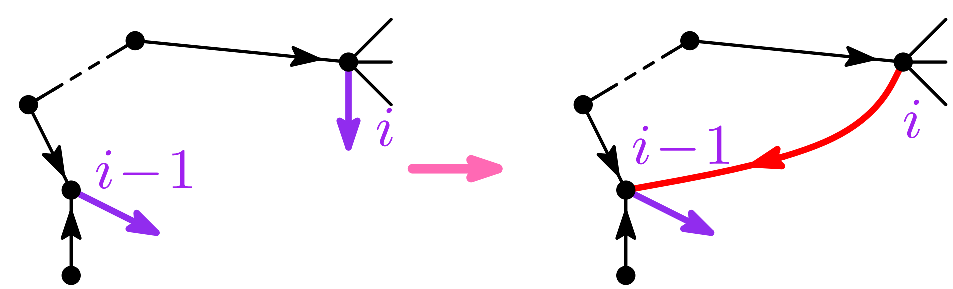
Unmatched stems = last 0, 1 and 2 corners

- Apply the following local rule :

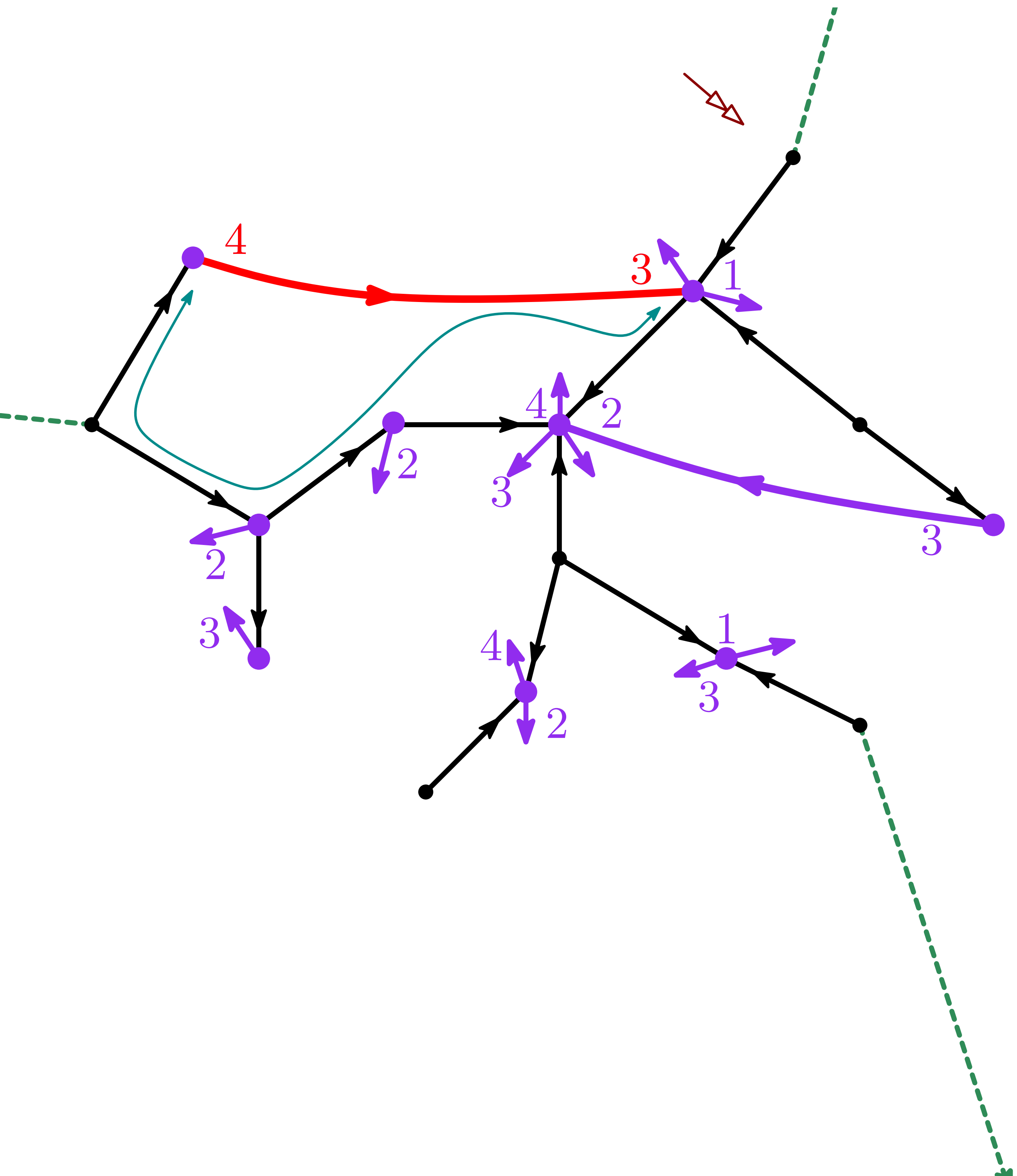


(= add a → before each descent and color the corresponding corner and vertex)

- Erase all non-purple and do the following closures



Same bijection with labels



- Label 0 the first corner.

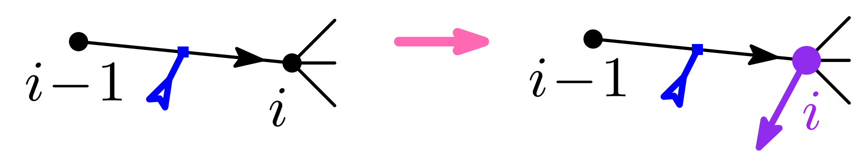
In clockwise order, apply the following rules:

- After an opening stem : increase by 1.
- After a closing stem : decrease by 1.

Labels ≈ depth of the face in the cubic map

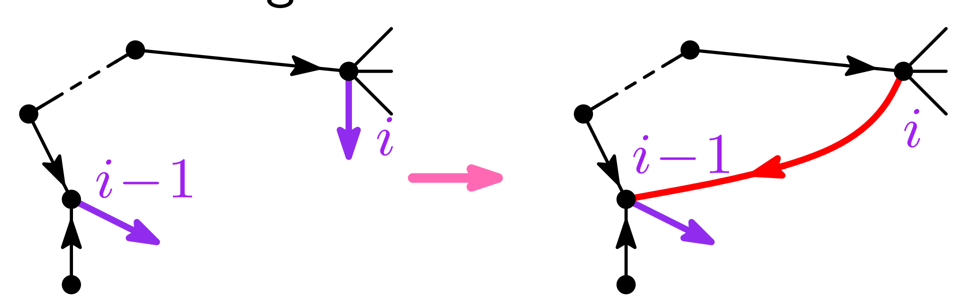
Unmatched stems = last 0, 1 and 2 corners

- Apply the following local rule :

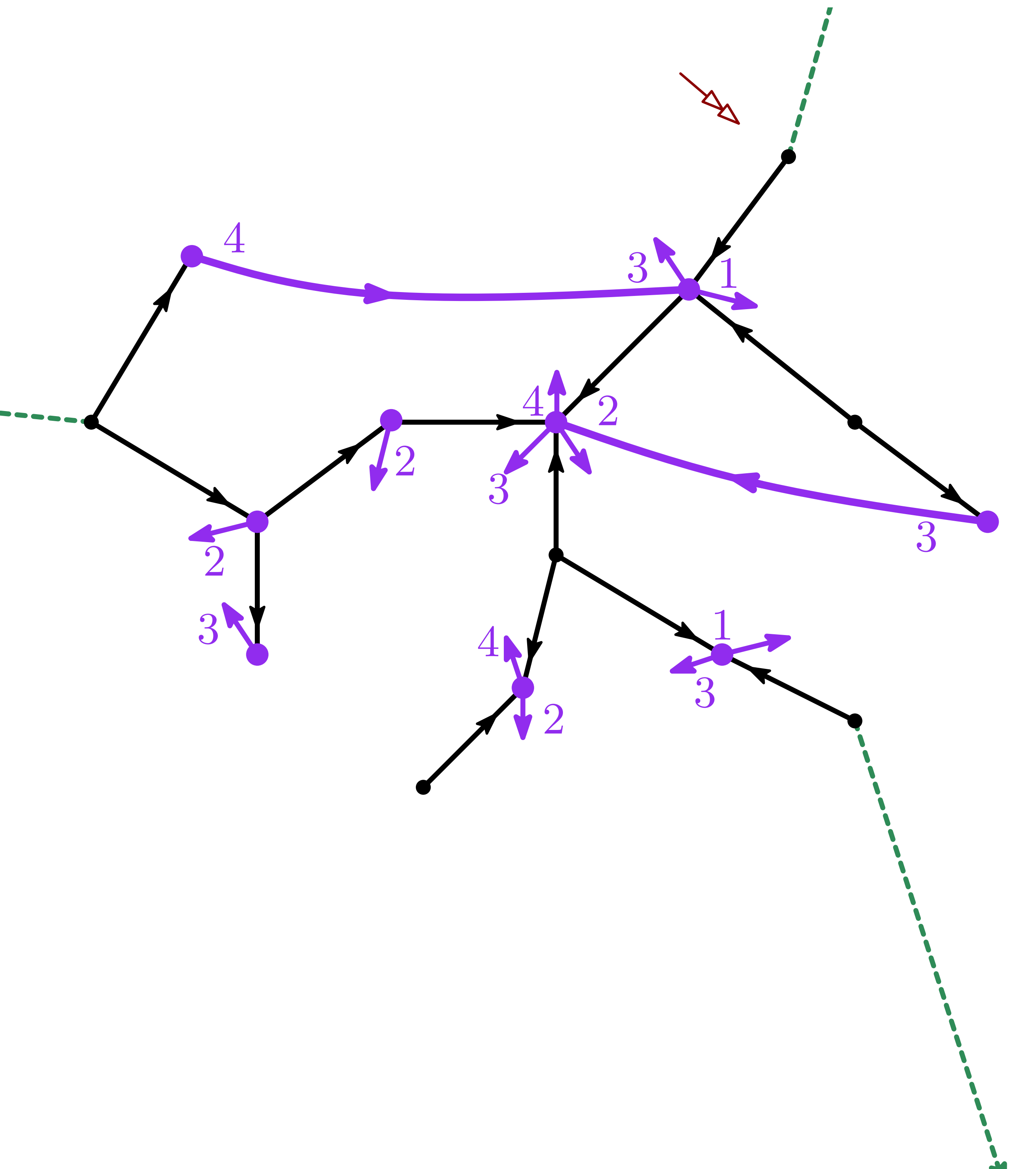


(= add a → before each descent and color the corresponding corner and vertex)

- Erase all non-purple and do the following closures



Same bijection with labels



- Label 0 the first corner.

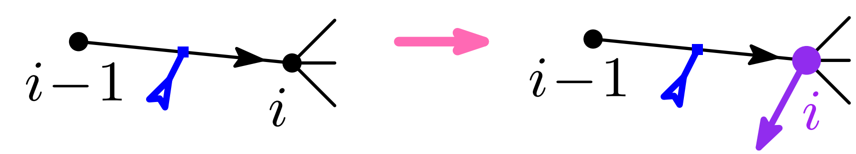
In clockwise order, apply the following rules:

- After an opening stem : increase by 1.
- After a closing stem : decrease by 1.

Labels $\approx$ depth of the face in the cubic map

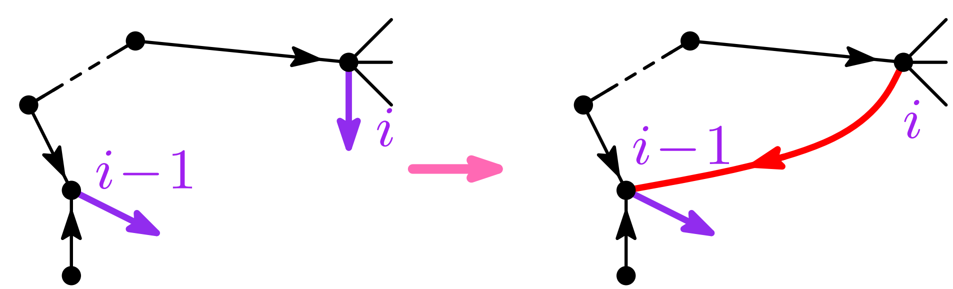
Unmatched stems = last 0, 1 and 2 corners

- Apply the following local rule :

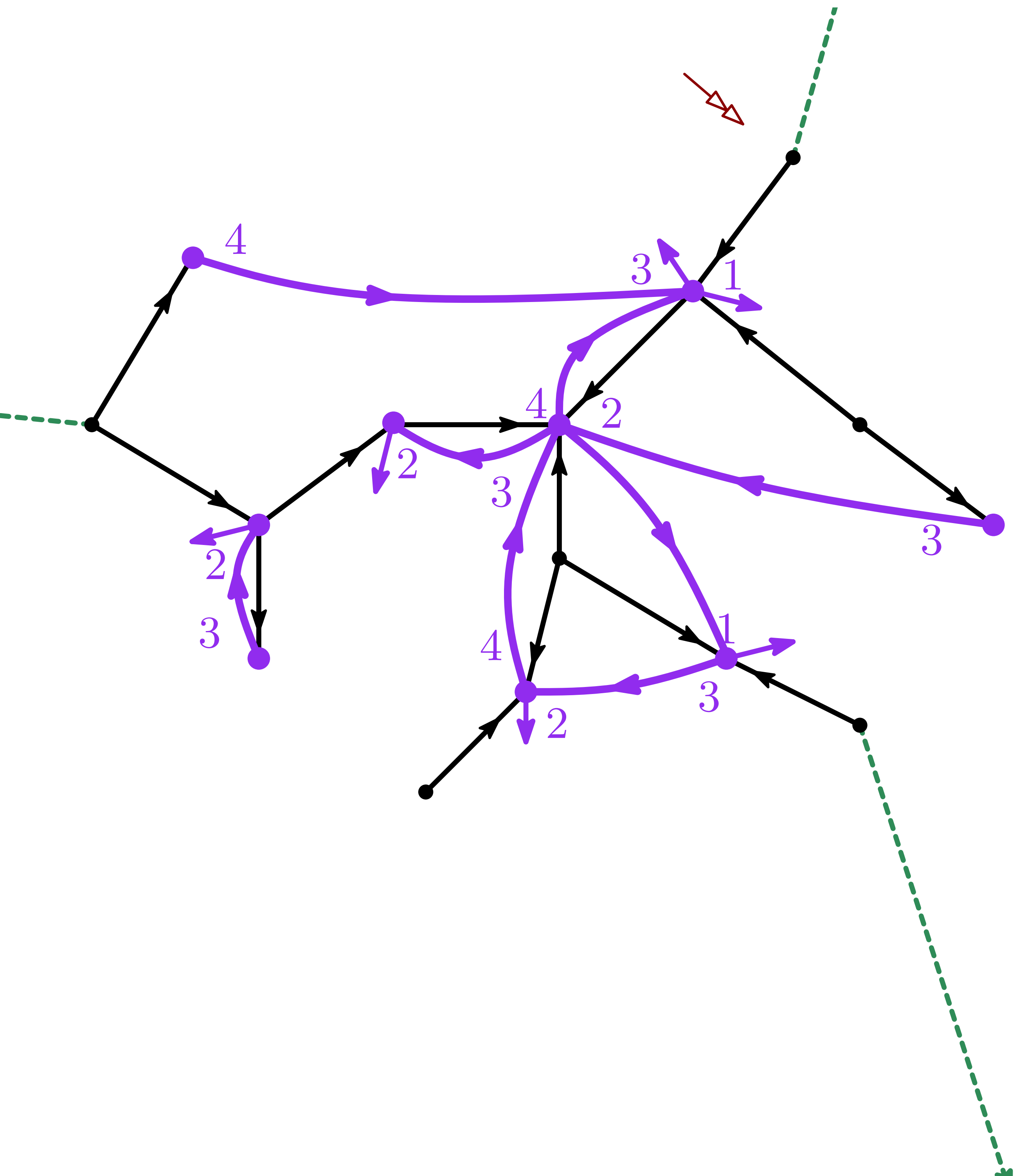


(= add a → before each descent and color the corresponding corner and vertex)

- Erase all non-purple and do the following closures



Same bijection with labels



- Label 0 the first corner.

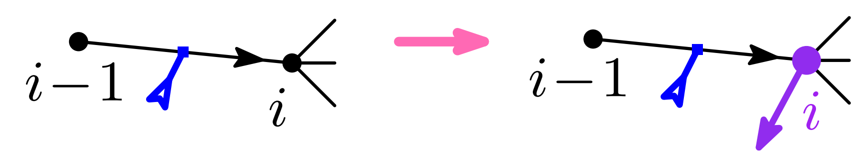
In clockwise order, apply the following rules:

- After an opening stem : increase by 1.
- After a closing stem : decrease by 1.

Labels ≈ depth of the face in the cubic map

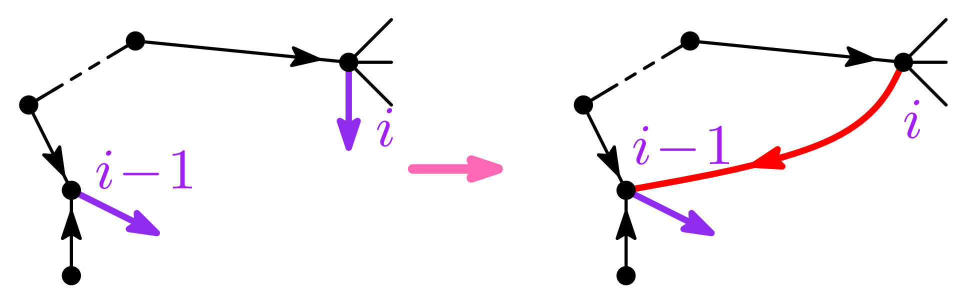
Unmatched stems = last 0, 1 and 2 corners

- Apply the following local rule :

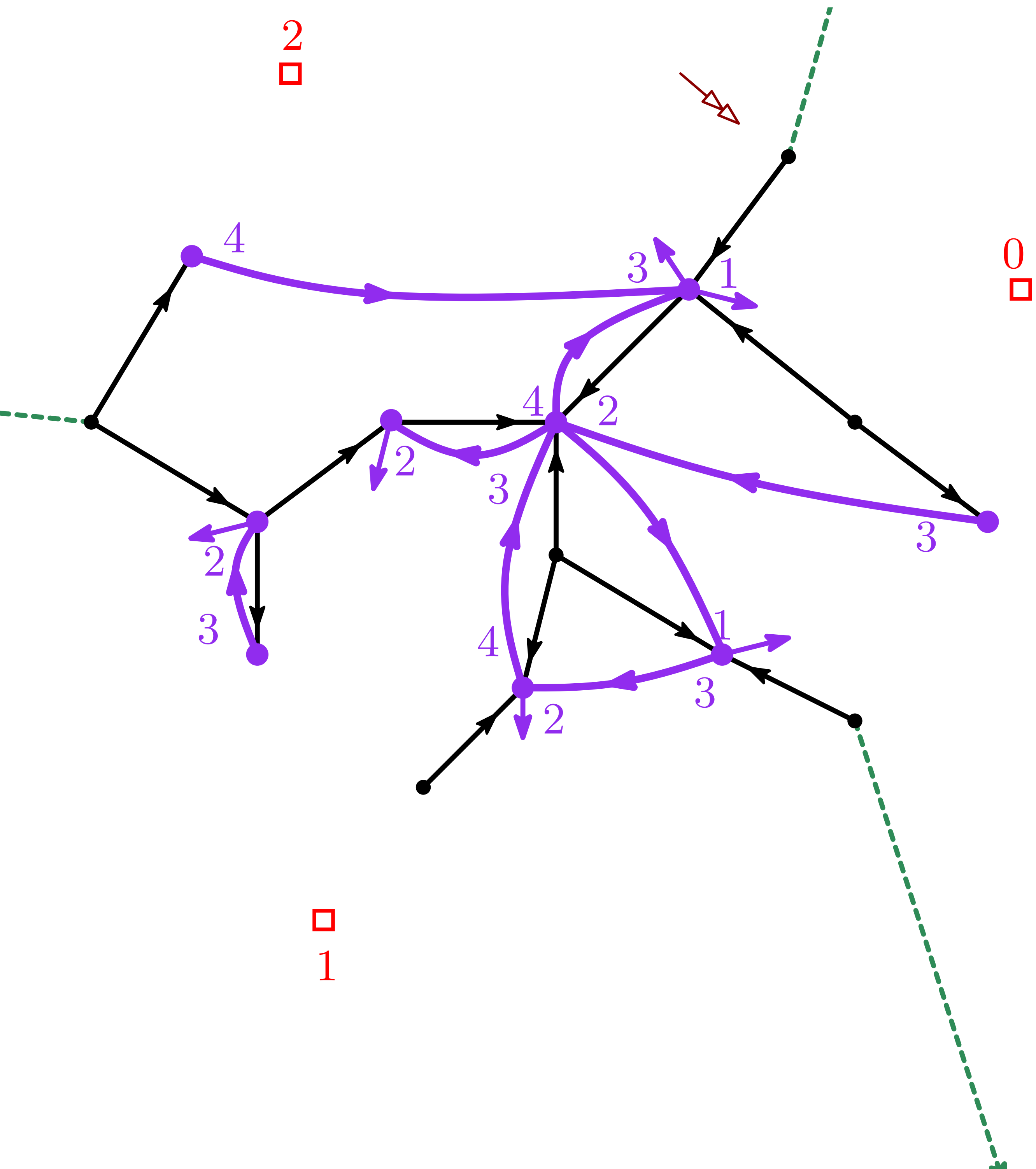


(= add a → before each descent and color the corresponding corner and vertex)

- Erase all non-purple and do the following closures



Same bijection with labels



- Label 0 the first corner.

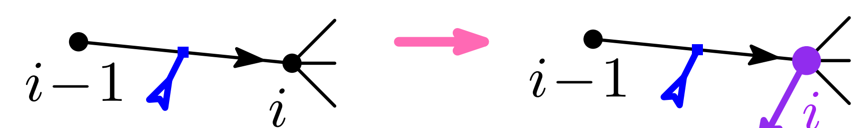
In clockwise order, apply the following rules:

- After an opening stem : increase by 1.
- After a closing stem : decrease by 1.

Labels ≈ depth of the face in the cubic map

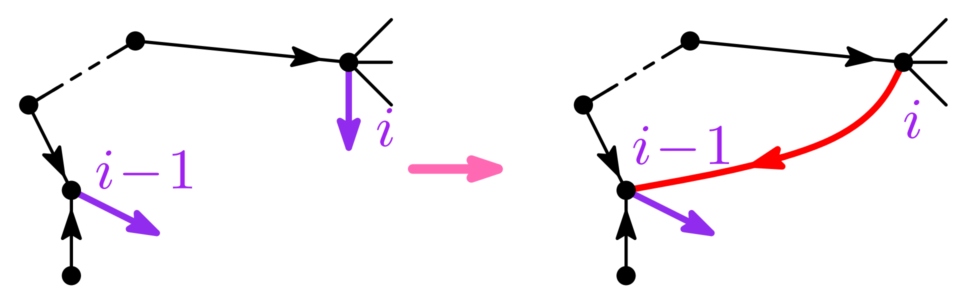
Unmatched stems = last 0, 1 and 2 corners

- Apply the following local rule :



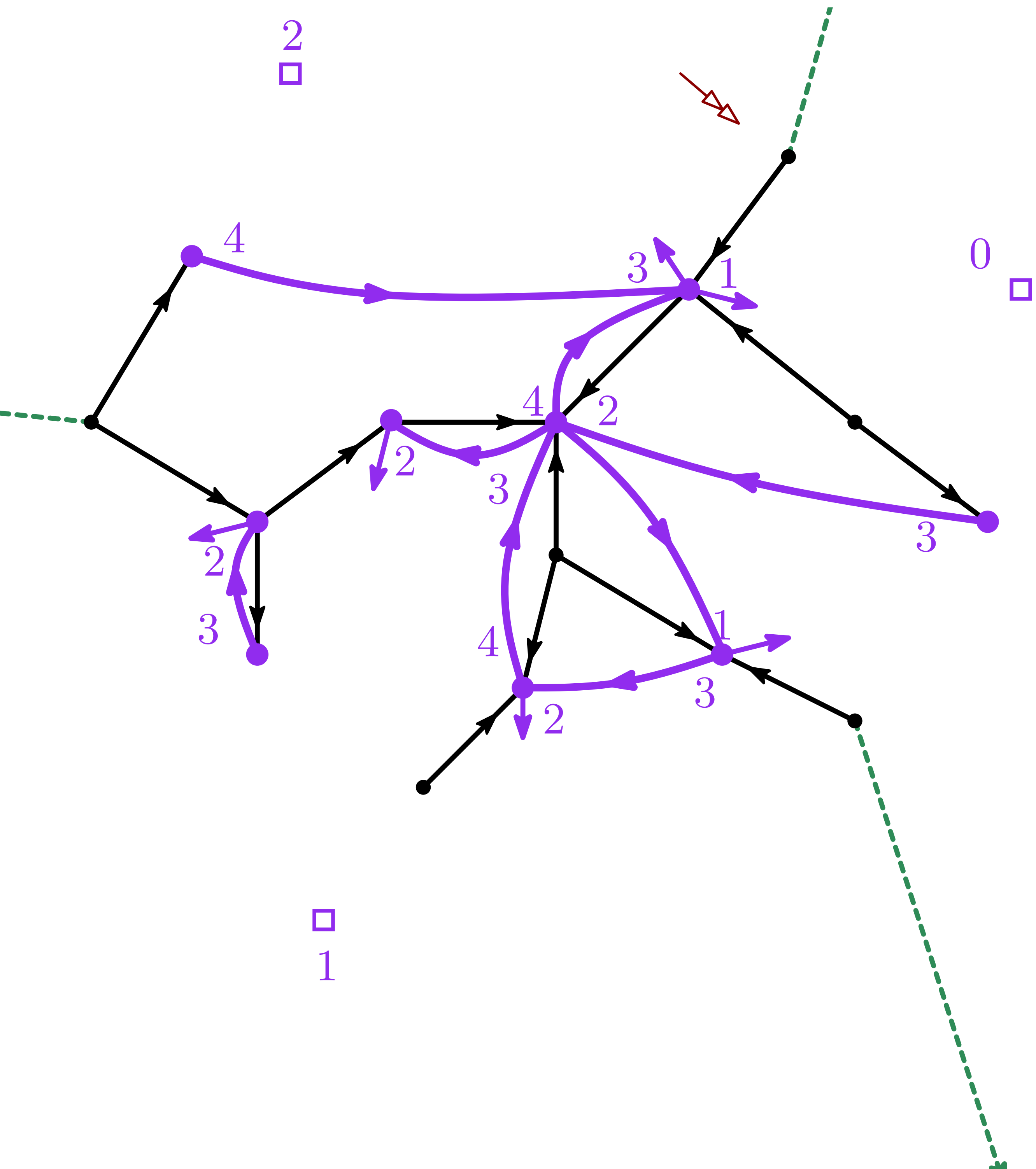
(= add a → before each descent and color the corresponding corner and vertex)

- Erase all non-purple and do the following closures



- Add 3 vertices and close the remaining opening stems in the same way.

Same bijection with labels



- Label 0 the first corner.

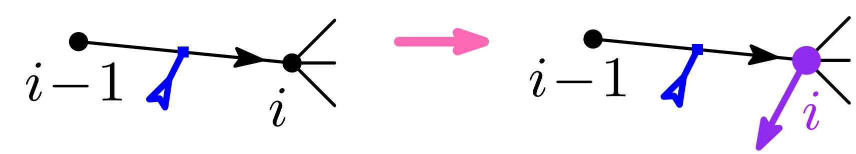
In clockwise order, apply the following rules:

- After an opening stem : increase by 1.
- After a closing stem : decrease by 1.

Labels $\approx$ depth of the face in the cubic map

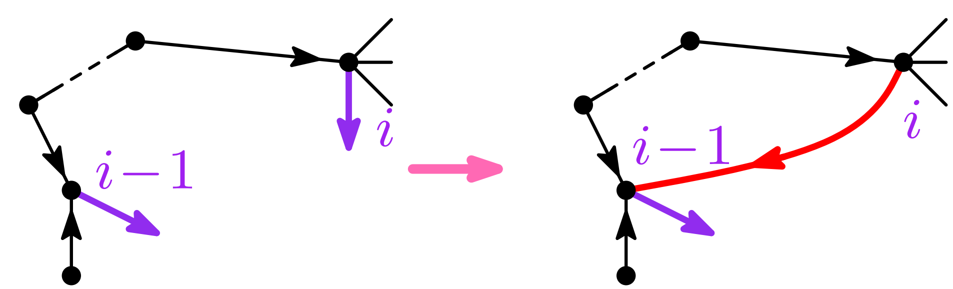
Unmatched stems = last 0, 1 and 2 corners

- Apply the following local rule :



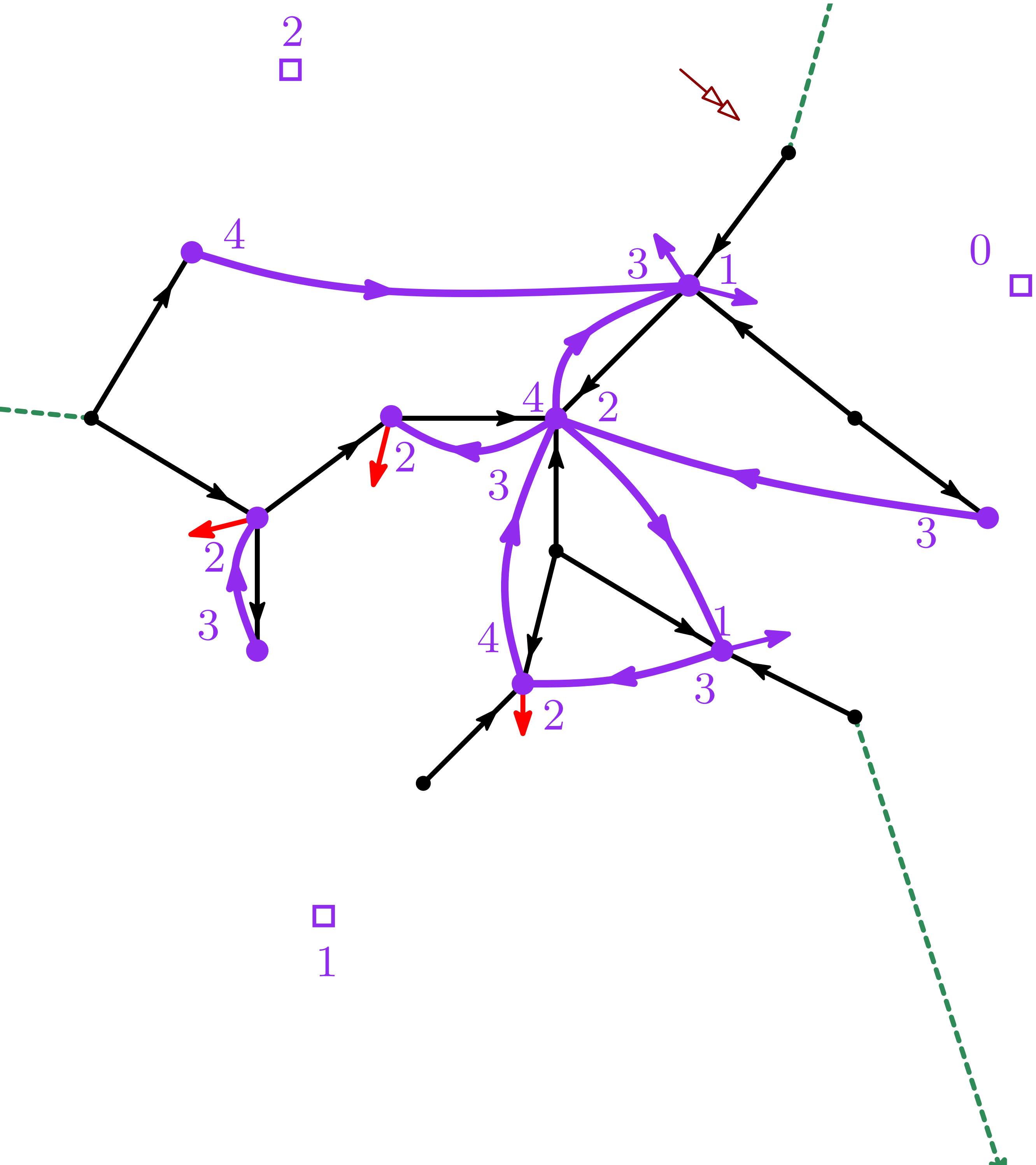
(= add a → before each descent and color the corresponding corner and vertex)

- Erase all non-purple and do the following closures



- Add 3 vertices and close the remaining opening stems in the same way.

Same bijection with labels



- Label 0 the first corner.

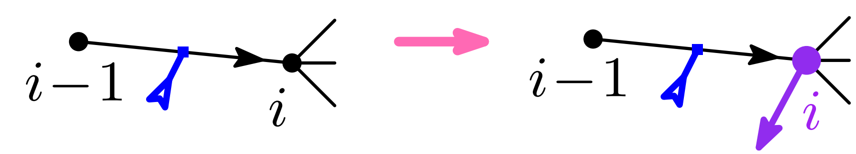
In clockwise order, apply the following rules:

- After an opening stem : increase by 1.
- After a closing stem : decrease by 1.

Labels $\approx$ depth of the face in the cubic map

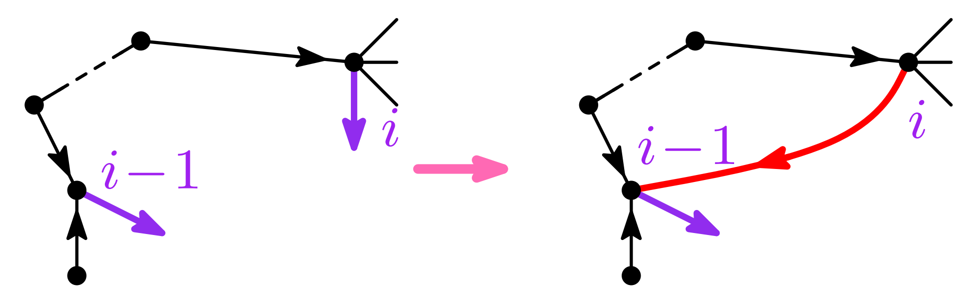
Unmatched stems = last 0, 1 and 2 corners

- Apply the following local rule :



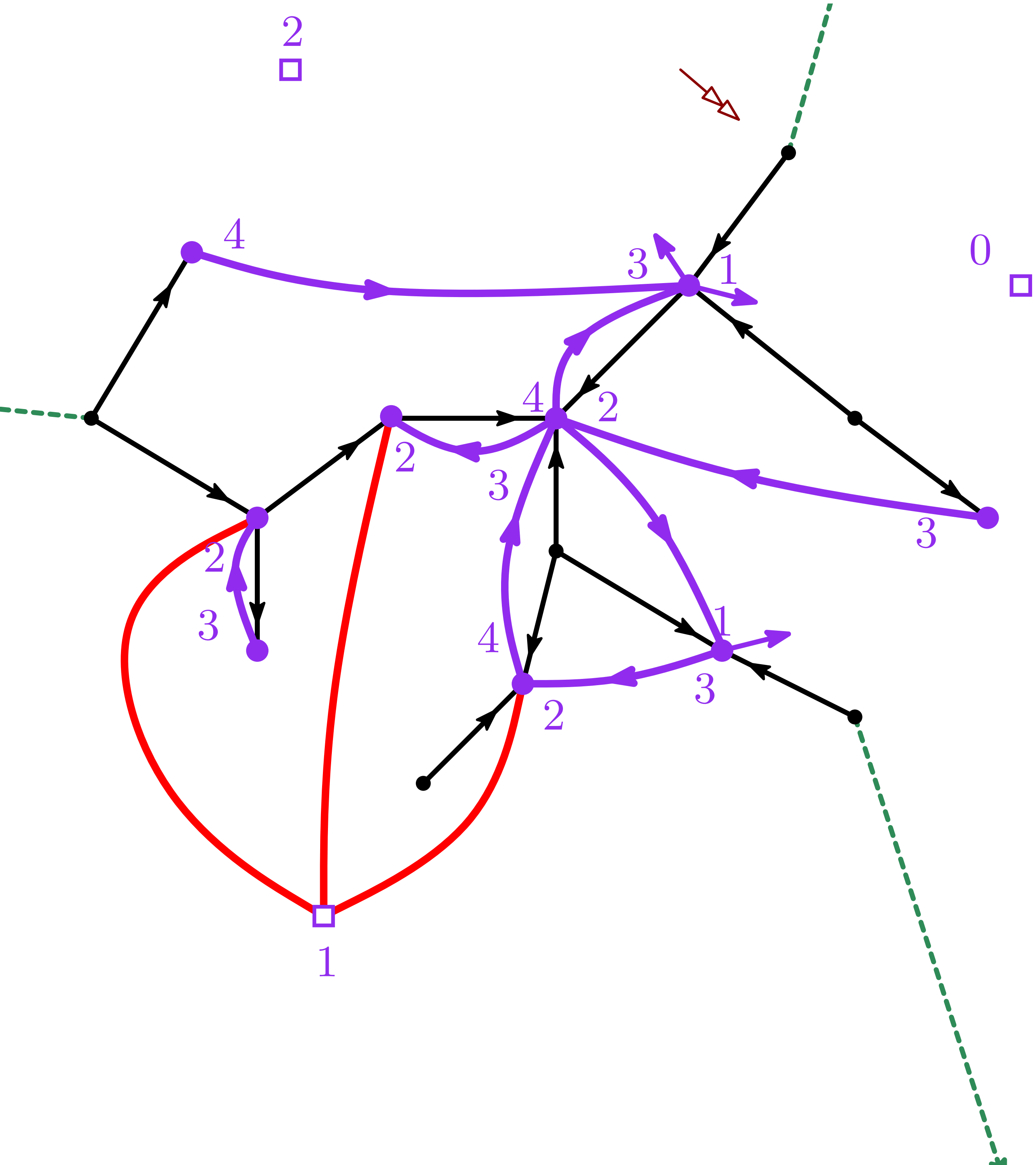
(= add a → before each descent and color the corresponding corner and vertex)

- Erase all non-purple and do the following closures



- Add 3 vertices and close the remaining opening stems in the same way.

Same bijection with labels



- Label 0 the first corner.

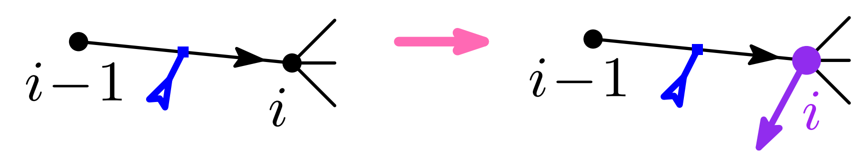
In clockwise order, apply the following rules:

- After an opening stem : increase by 1.
- After a closing stem : decrease by 1.

Labels ≈ depth of the face in the cubic map

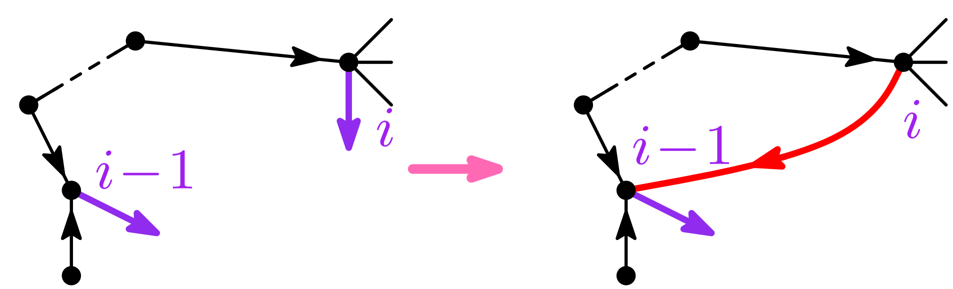
Unmatched stems = last 0, 1 and 2 corners

- Apply the following local rule :



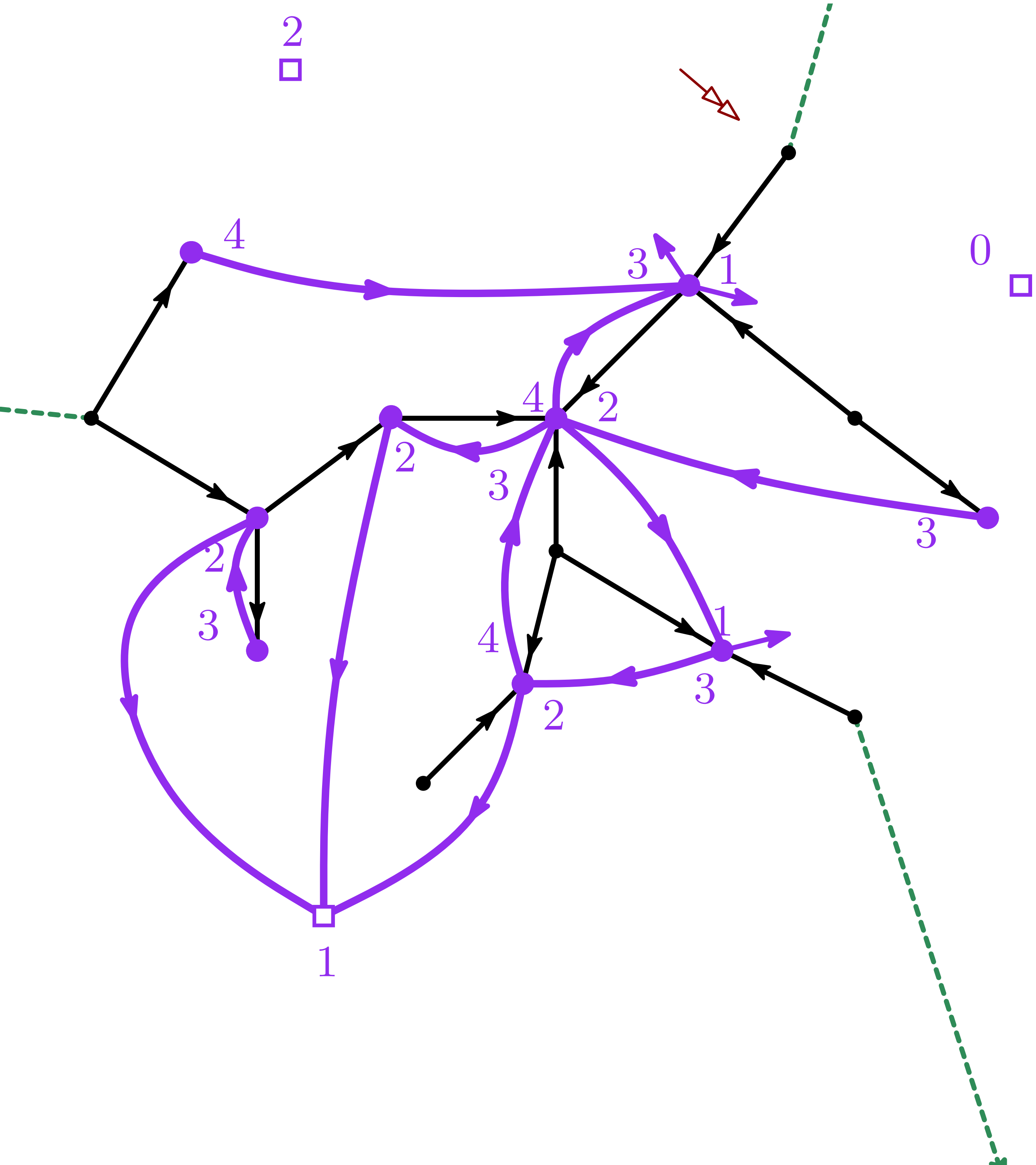
(= add a → before each descent and color the corresponding corner and vertex)

- Erase all non-purple and do the following closures



- Add 3 vertices and close the remaining opening stems in the same way.

Same bijection with labels



- Label 0 the first corner.

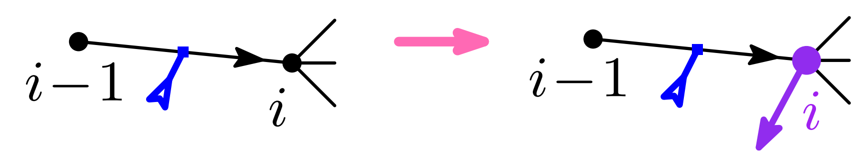
In clockwise order, apply the following rules:

- After an opening stem : increase by 1.
- After a closing stem : decrease by 1.

Labels $\approx$ depth of the face in the cubic map

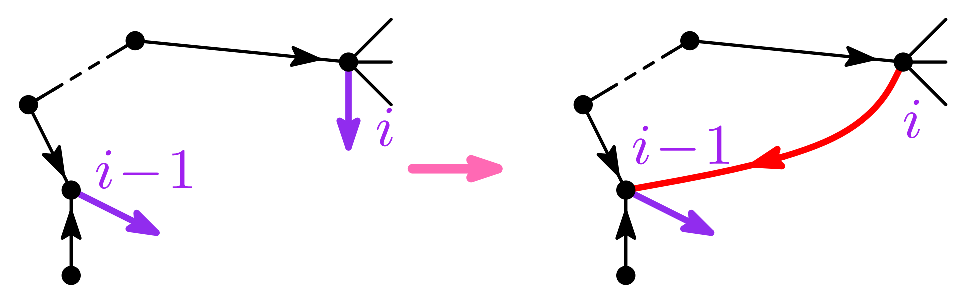
Unmatched stems = last 0, 1 and 2 corners

- Apply the following local rule :



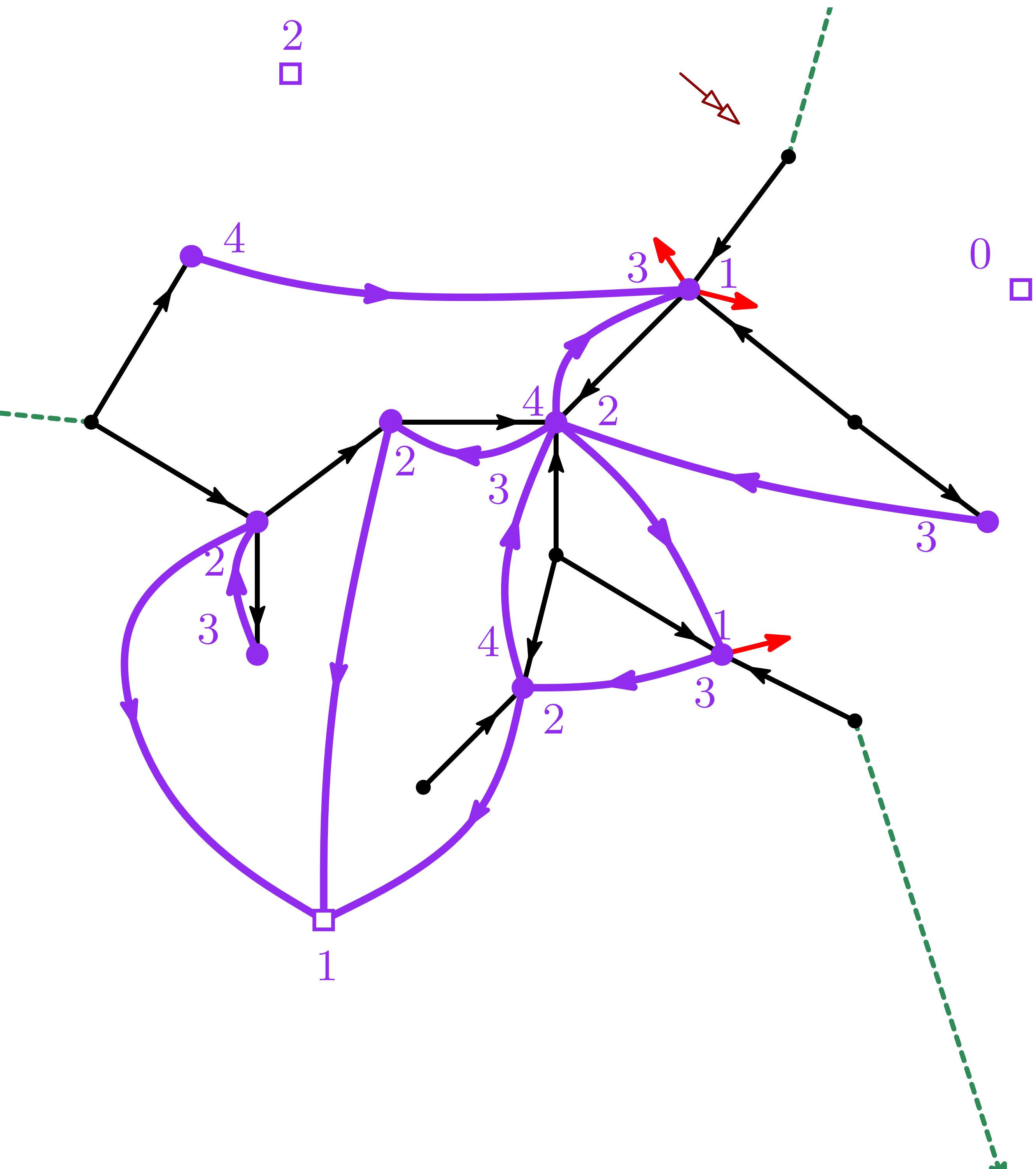
(= add a → before each descent and color the corresponding corner and vertex)

- Erase all non-purple and do the following closures



- Add 3 vertices and close the remaining opening stems in the same way.

Same bijection with labels



- Label 0 the first corner.

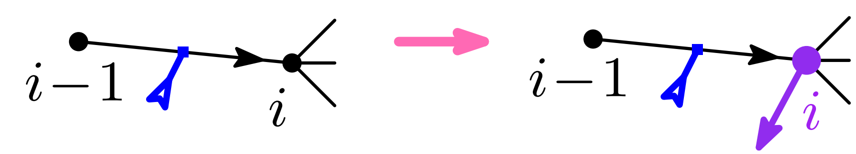
In clockwise order, apply the following rules:

- After an opening stem : increase by 1.
- After a closing stem : decrease by 1.

Labels ≈ depth of the face in the cubic map

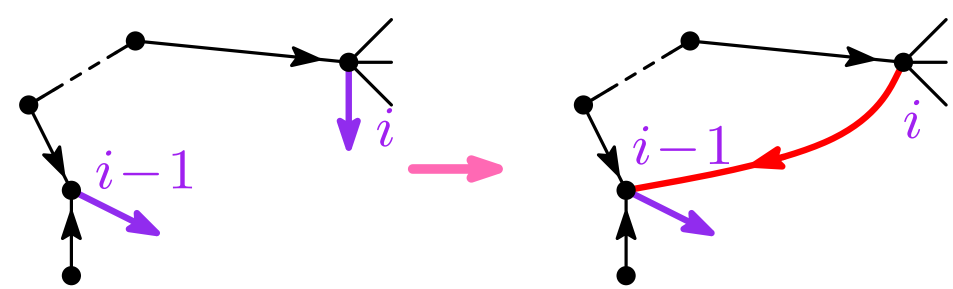
Unmatched stems = last 0, 1 and 2 corners

- Apply the following local rule :



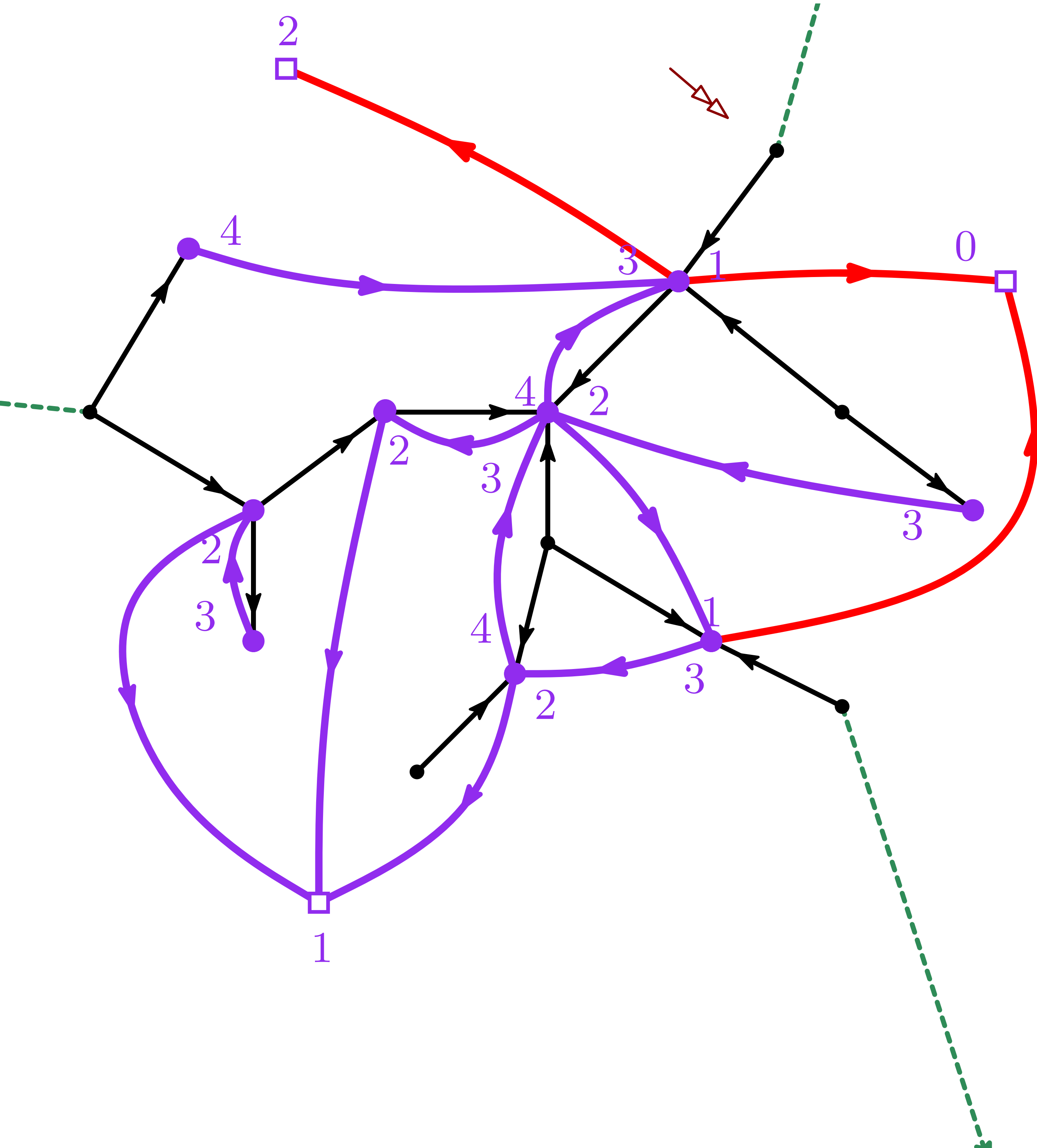
(= add a → before each descent and color the corresponding corner and vertex)

- Erase all non-purple and do the following closures



- Add 3 vertices and close the remaining opening stems in the same way.

Same bijection with labels



- Label 0 the first corner.

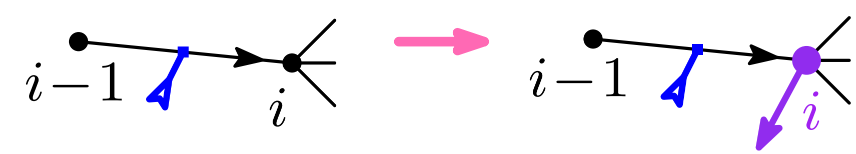
In clockwise order, apply the following rules:

- After an opening stem : increase by 1.
- After a closing stem : decrease by 1.

Labels ≈ depth of the face in the cubic map

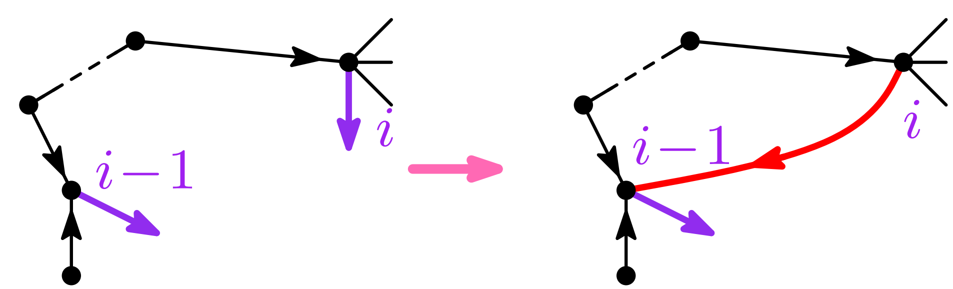
Unmatched stems = last 0, 1 and 2 corners

- Apply the following local rule :



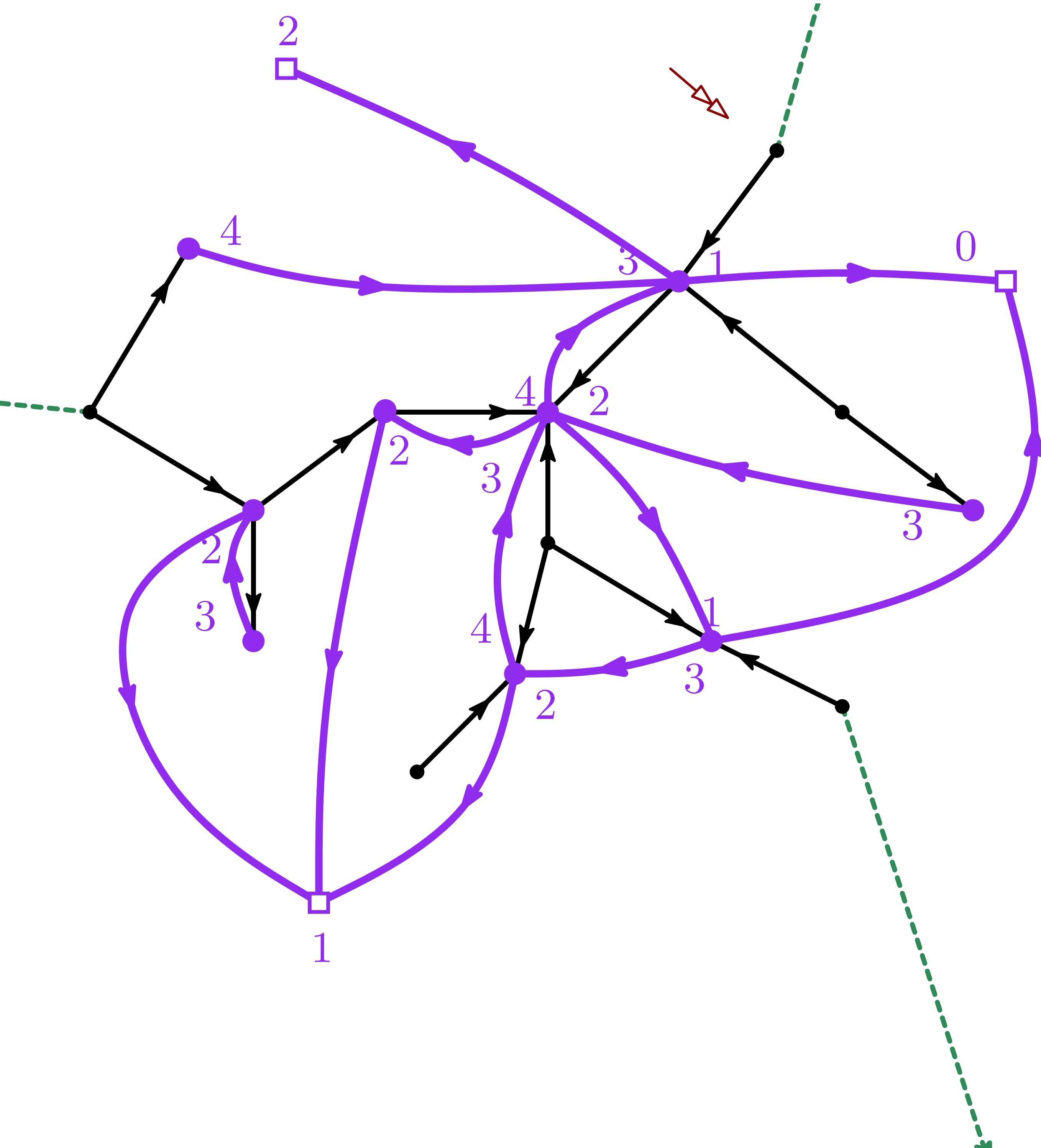
(= add a → before each descent and color the corresponding corner and vertex)

- Erase all non-purple and do the following closures



- Add 3 vertices and close the remaining opening stems in the same way.

Same bijection with labels



- Label 0 the first corner.

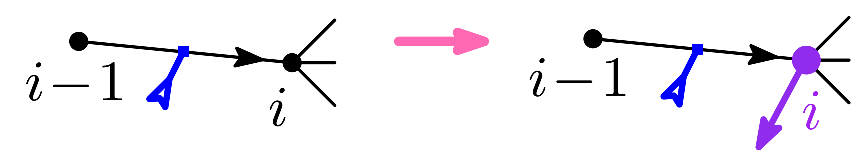
In clockwise order, apply the following rules:

- After an opening stem : increase by 1.
- After a closing stem : decrease by 1.

Labels ≈ depth of the face in the cubic map

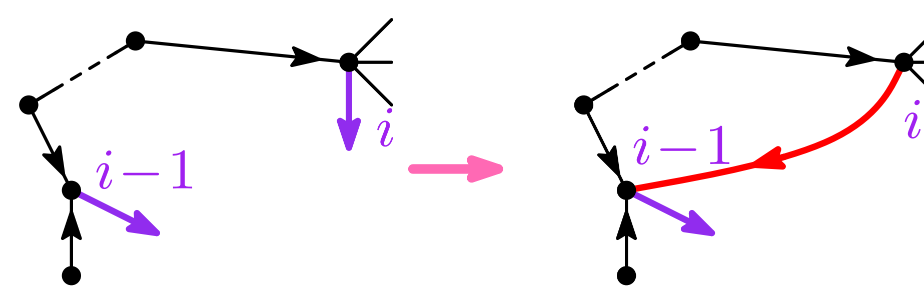
Unmatched stems = last 0, 1 and 2 corners

- Apply the following local rule :



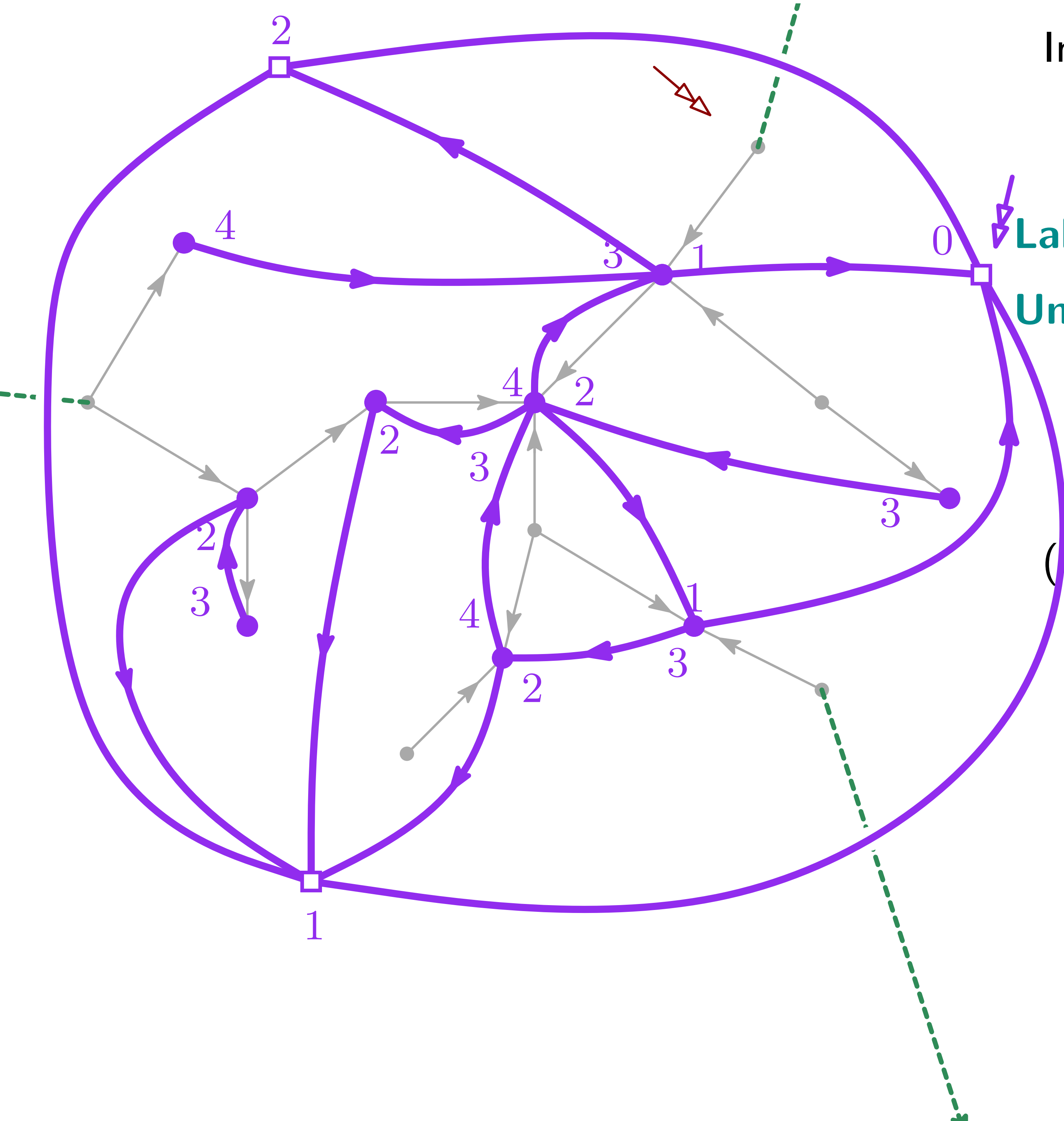
(= add a → before each descent and color the corresponding corner and vertex)

- Erase all non-purple and do the following closures



- Add 3 vertices and close the remaining opening stems in the same way.

Same bijection with labels



- Label 0 the first corner.

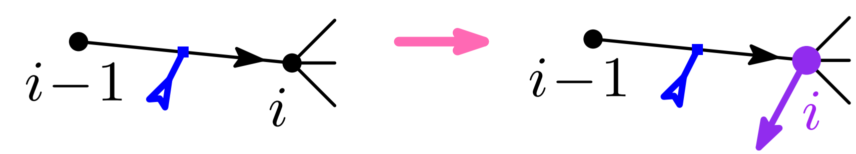
In clockwise order, apply the following rules:

- After an opening stem : increase by 1.
- After a closing stem : decrease by 1.

Labels ≈ depth of the face in the cubic map

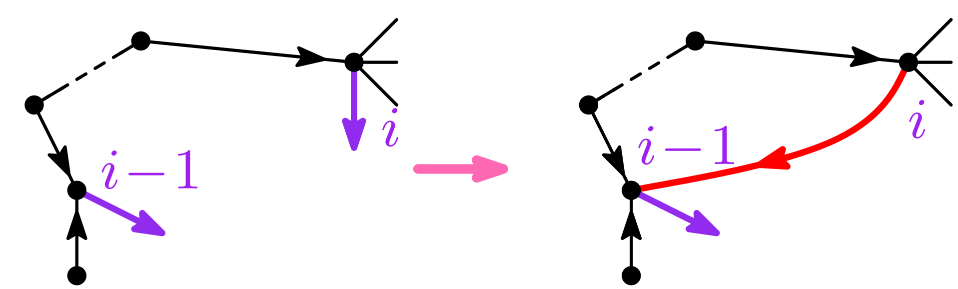
Unmatched stems = last 0, 1 and 2 corners

- Apply the following local rule :



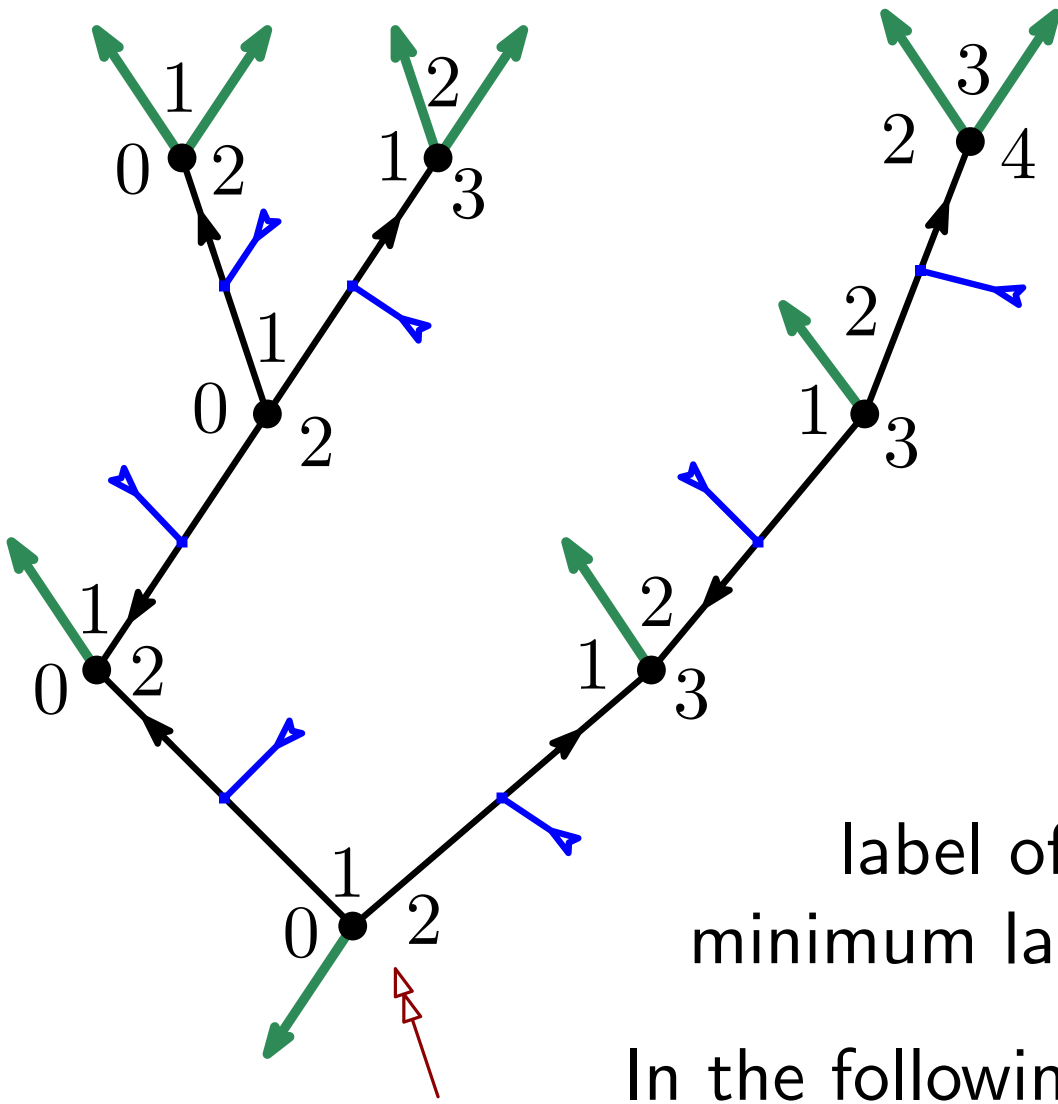
(= add a → before each descent and color the corresponding corner and vertex)

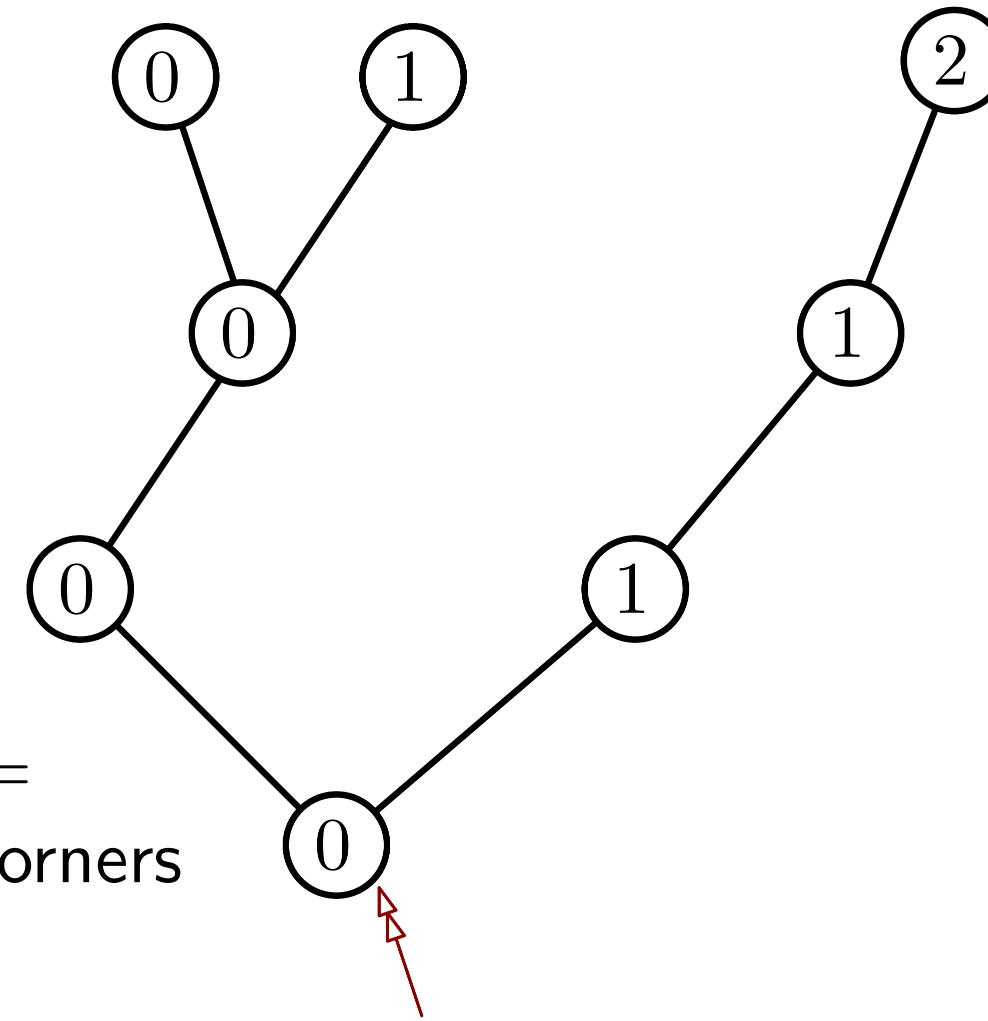
- Erase all non-purple and do the following closures



- Add 3 vertices and close the remaining opening stems in the same way.

From blossoming trees to labeled trees

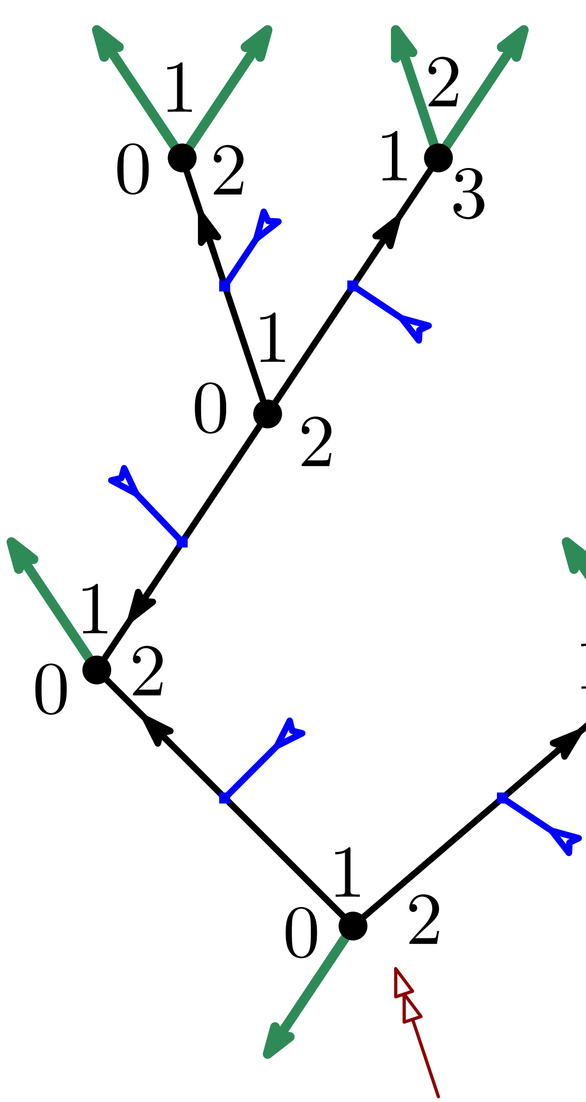




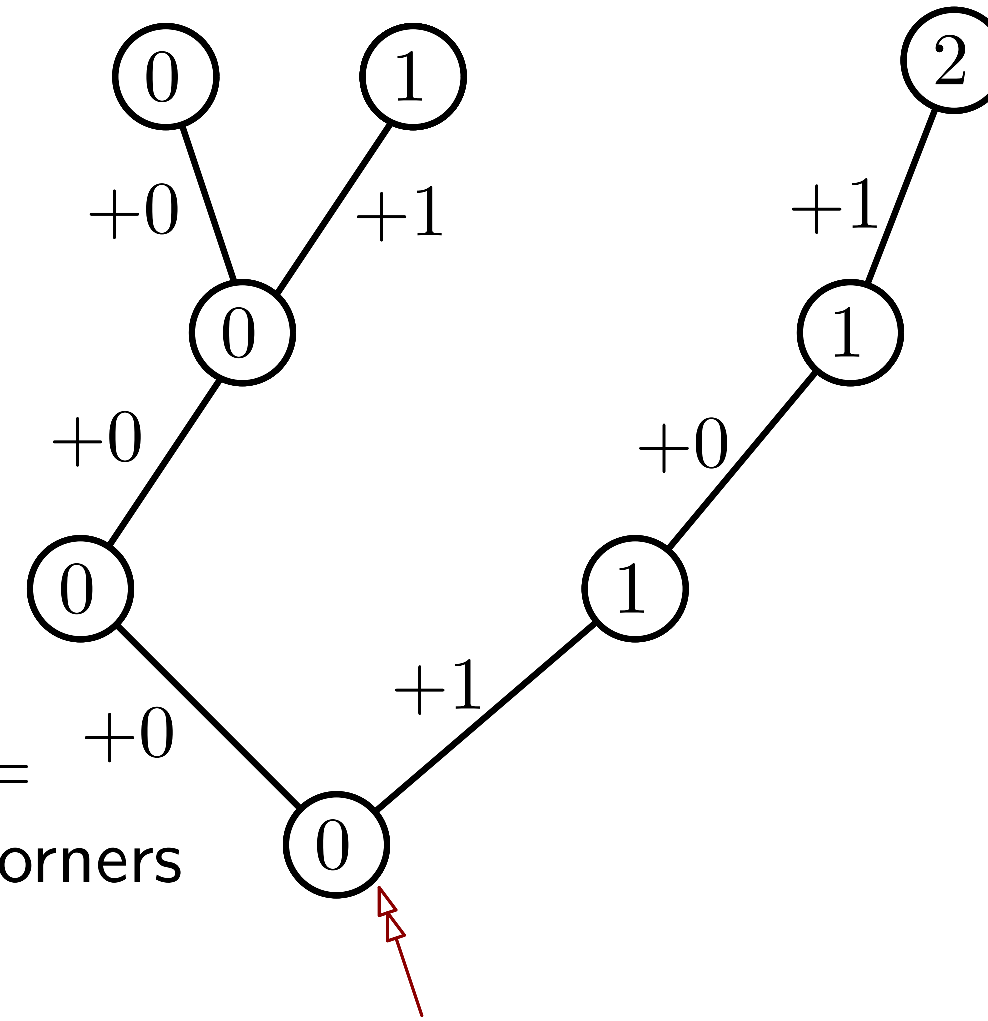
label of a vertex =
minimum label of its corners

In the following:
Labels gives approximate
distances to the root **in the map**

From blossoming trees to labeled trees



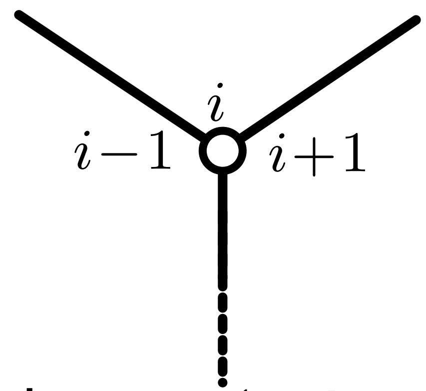
label of a vertex = minimum label of its corners



In the following:
Labels gives approximate distances to the root **in the map**

From blossoming trees to labeled trees

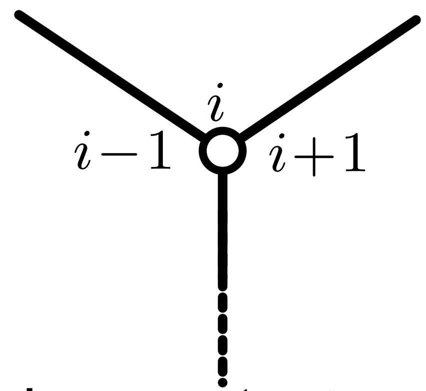
Around each vertex :



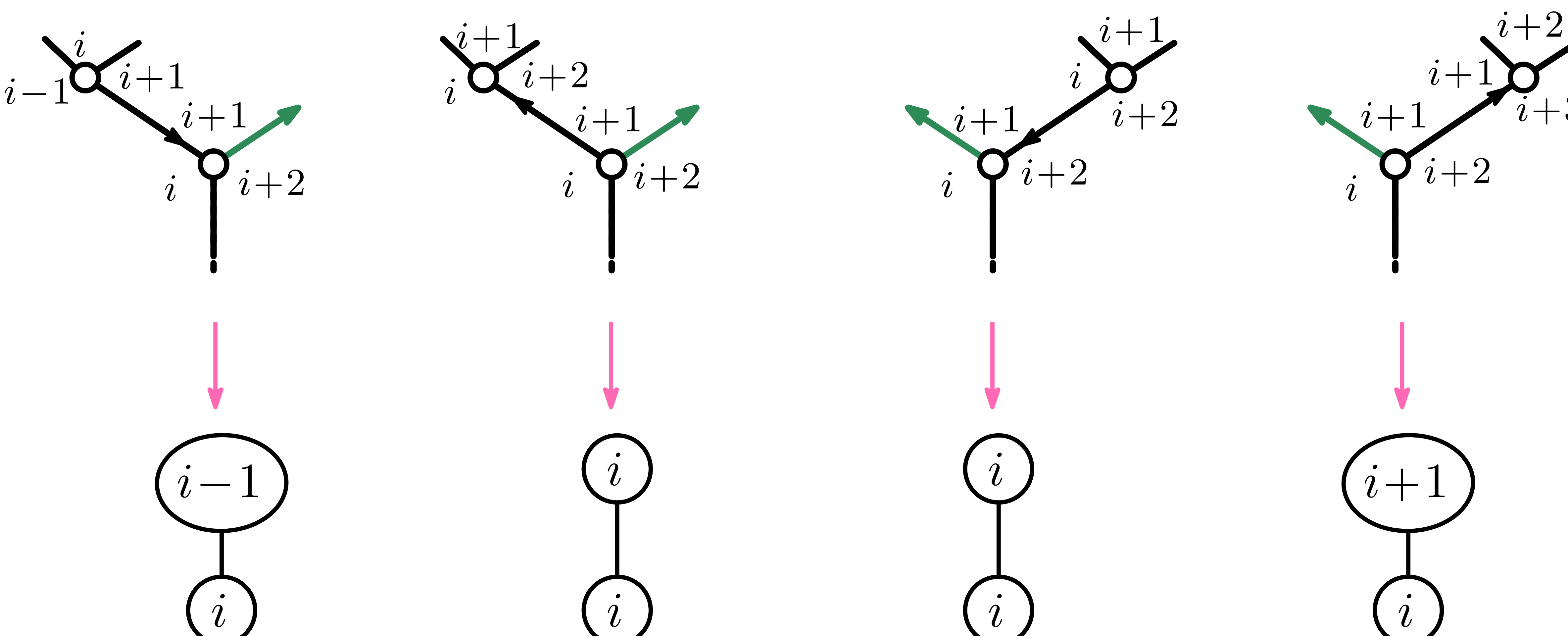
For instance, for a node of degree 1, 4 possibilities:

From blossoming trees to labeled trees

Around each vertex :



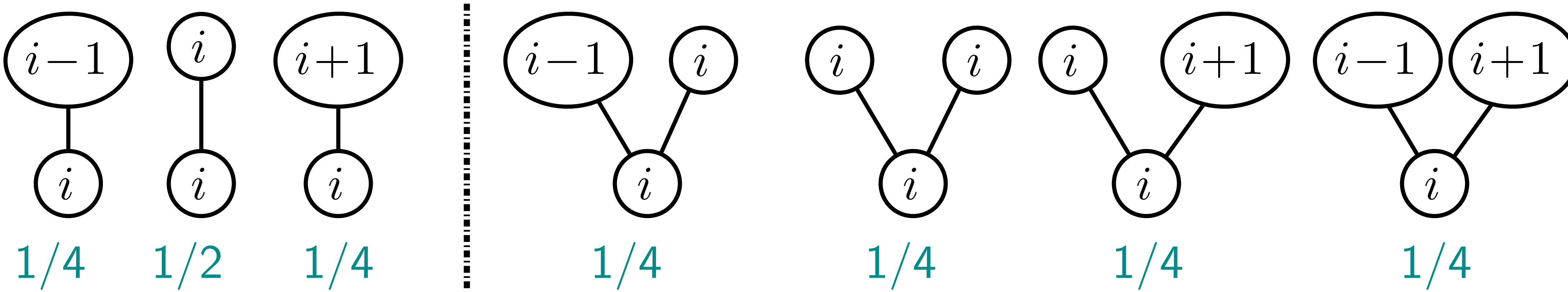
For instance, for a node of degree 1, 4 possibilities:



From blossoming trees to labeled trees

To do that :
- encode the maps by some trees.
- **study the limits of trees**,
- interpret the distance in the maps by some function of the tree.

- Labeled tree = GW binary trees + random displacements on edges



exactly the setting of [Marckert '08]:
convergence to the Brownian snake with the labels normalized by $(2n)^{1/4}$

Convergence of labeled trees

Theorem : [Marckert '08]
For a sequence of simple random outer-triangular maps (M_n), the contour and label process of the associated labeled tree satisfie:

$$\left((8n)^{-1/2}C_{\lfloor nt \rfloor}, (1/2n)^{1/4}\tilde{Z}_{\lfloor nt \rfloor}\right)_{0\le t\le 1} \xrightarrow[n\to\infty]{(d)} (e_t, Z_t)_{0\le t\le 1},$$

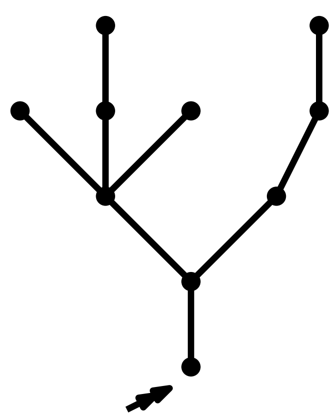
Contour and label processes of a labeled tree

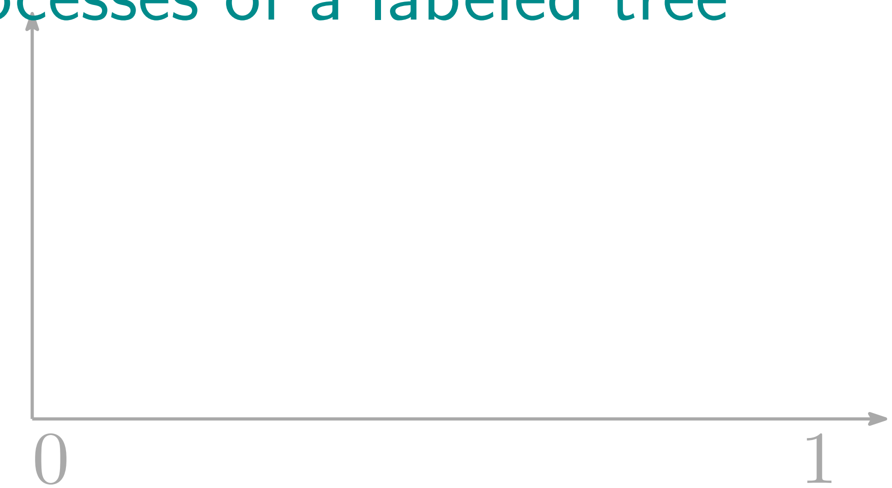
Convergence of labeled trees

Theorem : [Marckert '08]
For a sequence of simple random outer-triangular maps (M_n), the contour and label process of the associated labeled tree satisfie:

$$\left((8n)^{-1/2} C_{\lfloor nt \rfloor}, (1/2n)^{1/4} \tilde{Z}_{\lfloor nt \rfloor}\right)_{0 \leq t \leq 1} \underset{n \to \infty}{\overset{(d)}{\to}} (e_t, Z_t)_{0 \leq t \leq 1},$$

Contour and label processes of a labeled tree



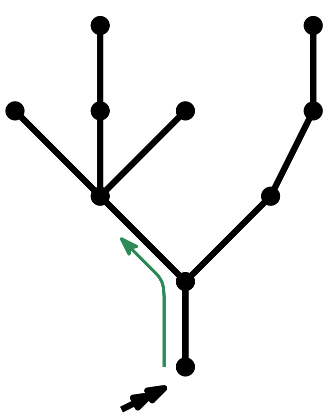


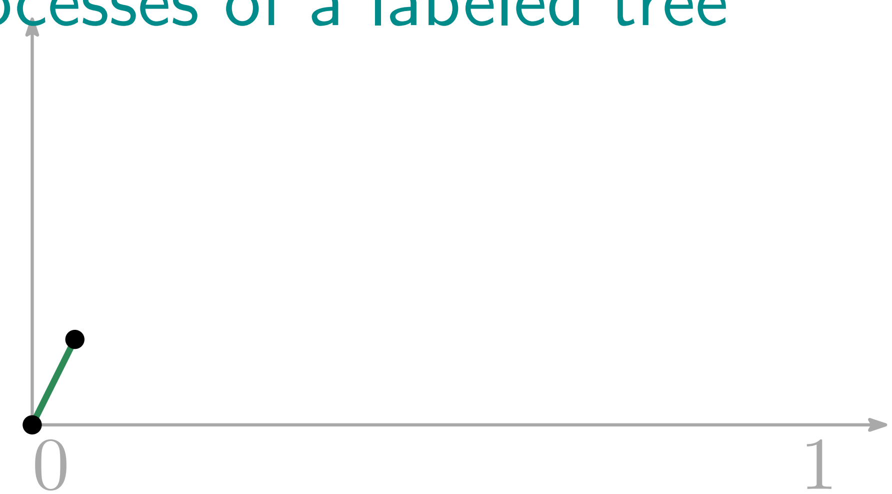
Convergence of labeled trees

Theorem : [Marckert '08]
For a sequence of simple random outer-triangular maps (M_n), the contour and label process of the associated labeled tree satisfie:

$$\left((8n)^{-1/2}C_{\lfloor nt\rfloor},(1/2n)^{1/4}\tilde{Z}_{\lfloor nt\rfloor}\right)_{0\leq t\leq 1}\ \underset{n\to\infty}{\overset{(d)}{\to}}\ (e_t,Z_t)_{0\leq t\leq 1},$$

Contour and label processes of a labeled tree



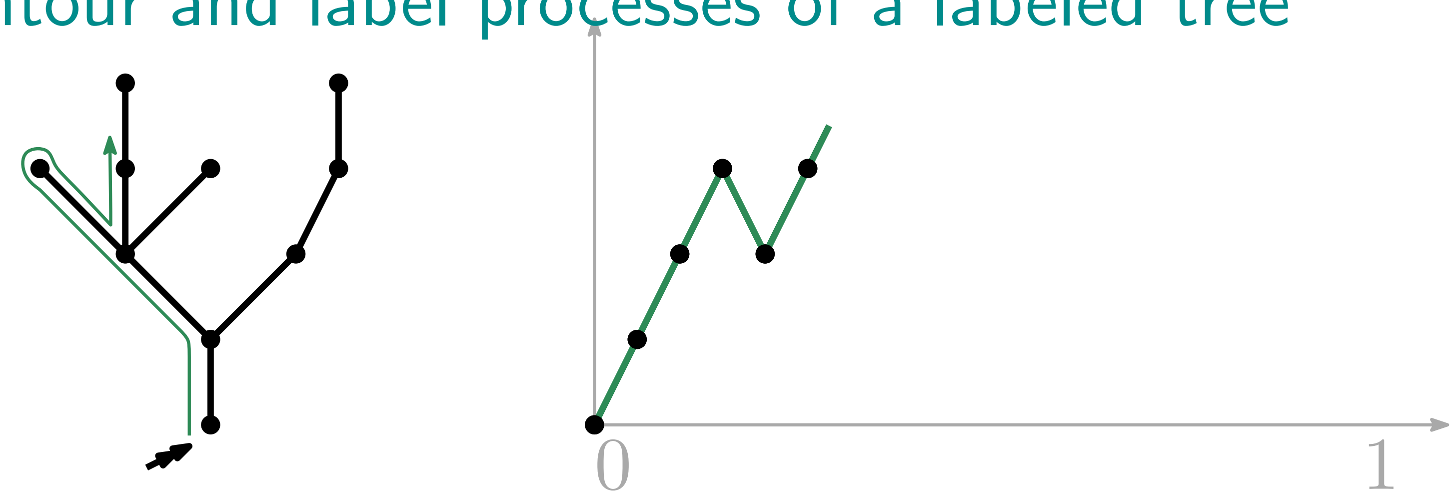


Convergence of labeled trees

Theorem : [Marckert '08]
For a sequence of simple random outer-triangular maps (M_n), the contour and label process of the associated labeled tree satisfie:

$$\left((8n)^{-1/2}C_{\lfloor nt \rfloor}, (1/2n)^{1/4}\tilde{Z}_{\lfloor nt \rfloor}\right)_{0\leq t\leq 1} \overset{(d)}{\underset{n\to\infty}{\to}} (e_t, Z_t)_{0\leq t\leq 1},$$

Contour and label processes of a labeled tree

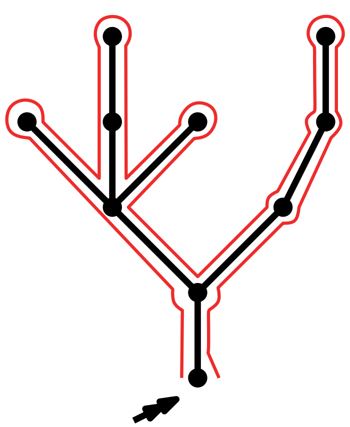


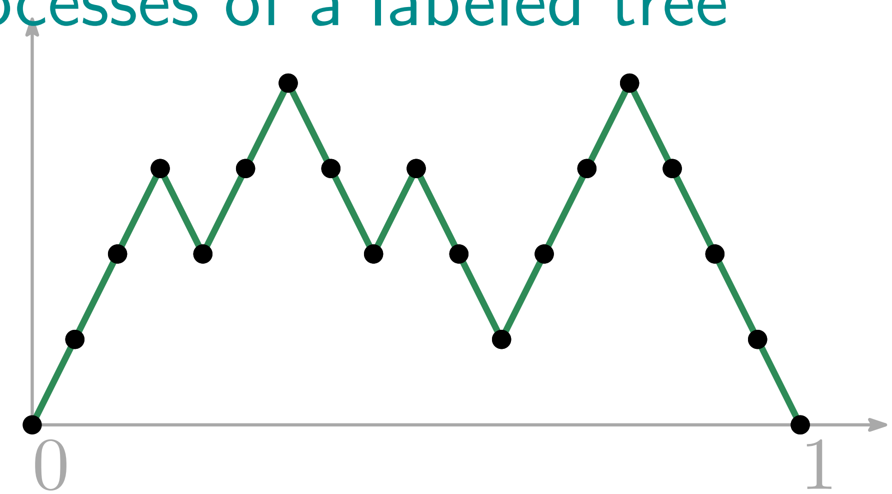
Convergence of labeled trees

Theorem : [Marckert '08]
For a sequence of simple random outer-triangular maps (M_n), the contour and label process of the associated labeled tree satisfie:

$$\left((8n)^{-1/2}C_{\lfloor nt\rfloor},(1/2n)^{1/4}\tilde{Z}_{\lfloor nt\rfloor}\right)_{0\le t\le 1}\ \overset{(d)}{\underset{n\to\infty}{\longrightarrow}}\ (e_t,Z_t)_{0\le t\le 1},$$

Contour and label processes of a labeled tree



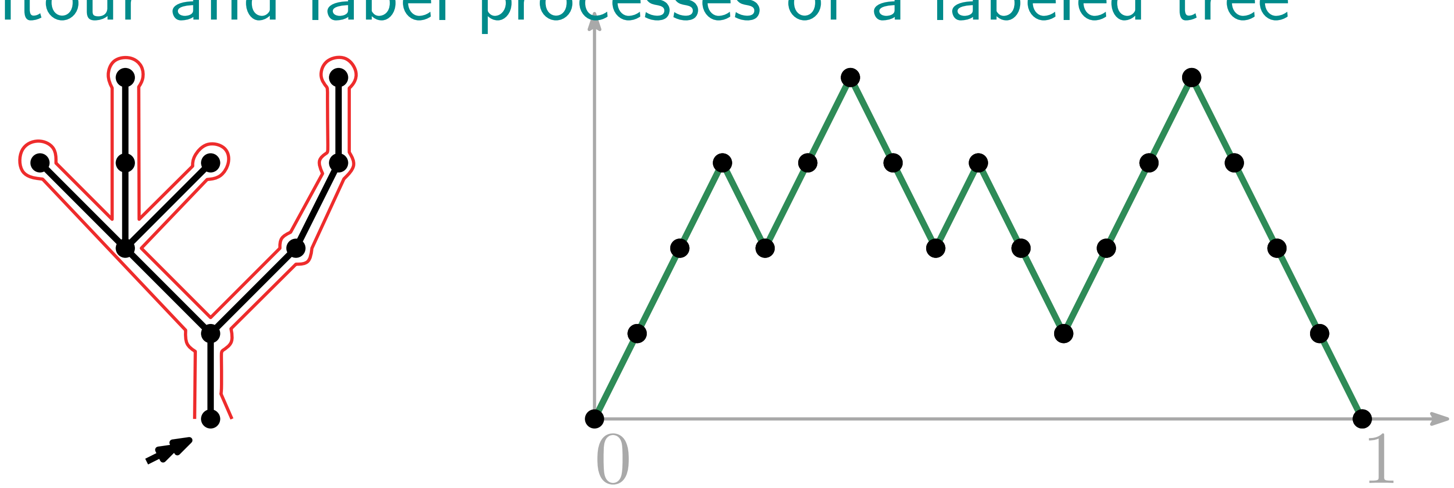


Convergence of labeled trees

Theorem : [Marckert '08]
For a sequence of simple random outer-triangular maps (M_n), the contour and label process of the associated labeled tree satisfie:

$$\left((8n)^{-1/2}C_{\lfloor nt \rfloor}, (1/2n)^{1/4}\tilde{Z}_{\lfloor nt \rfloor}\right)_{0\le t\le 1} \xrightarrow[n\to\infty]{(d)} (e_t, Z_t)_{0\le t\le 1},$$

Contour and label processes of a labeled tree



T ⟶ C_n^T (or C_n) = contour process

Convergence of labeled trees

Theorem : [Marckert '08]
For a sequence of simple random outer-triangular maps (M_n), the contour and label process of the associated labeled tree satisfie:

$$\left((8n)^{-1/2}C_{\lfloor nt\rfloor}, (1/2n)^{1/4}\tilde{Z}_{\lfloor nt\rfloor}\right)_{0\leq t\leq 1} \xrightarrow[n\to\infty]{(d)} (e_t, Z_t)_{0\leq t\leq 1},$$

Contour and label processes of a labeled tree

i and j = same vertex of T iff $C_n(i) = C_n(j) = \min_{i\leq k\leq j} C_n(k)$

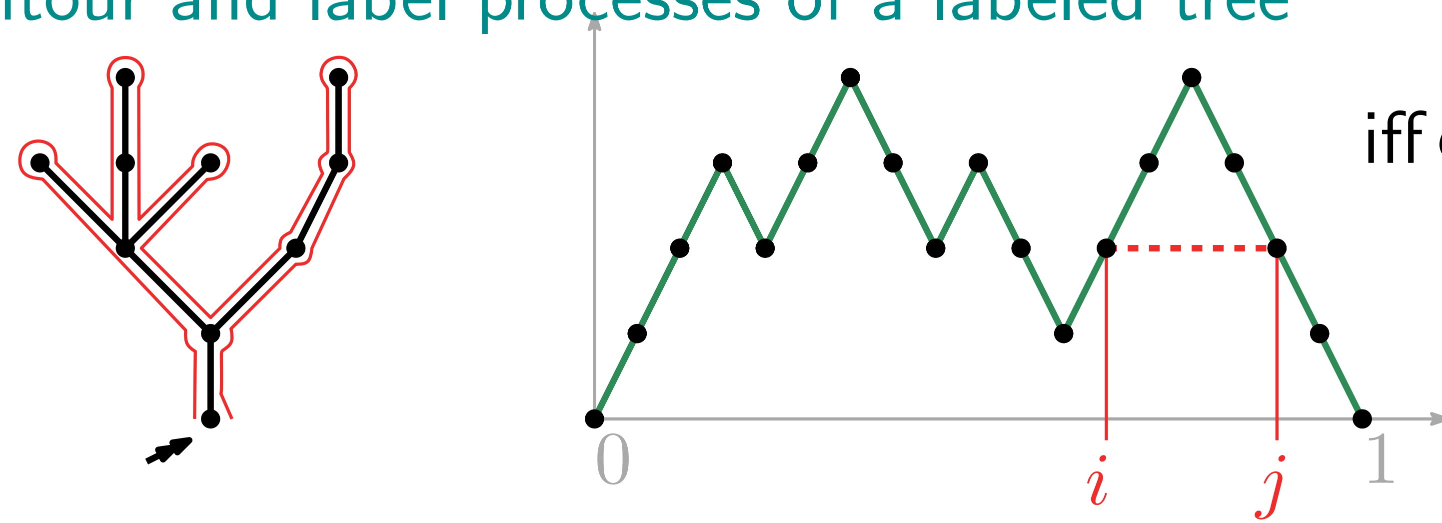
$T \longrightarrow C_n^T$ (or C_n) = contour process

Convergence of labeled trees

Theorem : [Marckert '08]
For a sequence of simple random outer-triangular maps (M_n), the contour and label process of the associated labeled tree satisfie:

$$\left((8n)^{-1/2}C_{\lfloor nt\rfloor},(1/2n)^{1/4}\tilde{Z}_{\lfloor nt\rfloor}\right)_{0\le t\le 1}\xrightarrow[n\to\infty]{(d)}(e_t,Z_t)_{0\le t\le 1},$$

Contour and label processes of a labeled tree



i and j = same vertex of T
iff $C_n(i)=C_n(j)=\min_{i\le k\le j} C_n(k)$

T ⟶ C_n^T (or C_n) = contour process

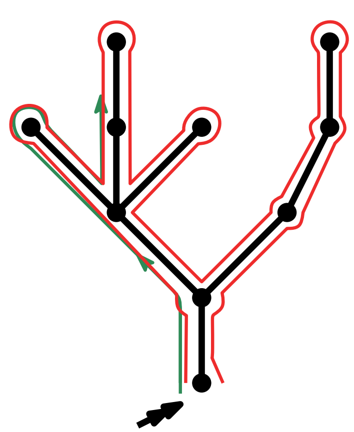
If T is a labeled tree, $(C_n(i), Z_n(i))$ = contour and label processes

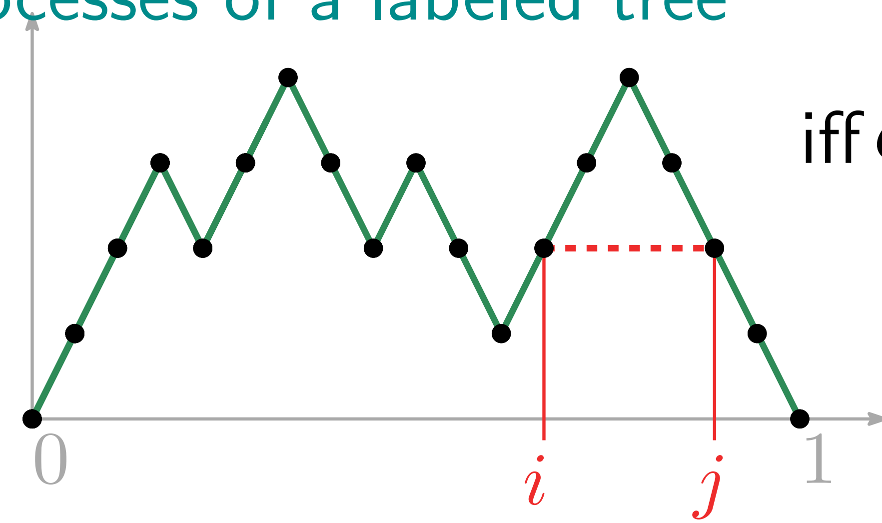
Convergence of labeled trees

Theorem : [Marckert '08]
For a sequence of simple random outer-triangular maps (M_n), the contour and label process of the associated labeled tree satisfie:

$$\left((8n)^{-1/2}C_{\lfloor nt\rfloor},(1/2n)^{1/4}\tilde{Z}_{\lfloor nt\rfloor}\right)_{0\leq t\leq 1}\ \underset{n\to\infty}{\overset{(d)}{\rightarrow}}\ (e_t,Z_t)_{0\leq t\leq 1},$$

Contour and label processes of a labeled tree





i and j = same vertex of T
iff $C_n(i)=C_n(j)=\min_{i\leq k\leq j} C_n(k)$

T ⟶ C_n^T (or C_n) = contour process

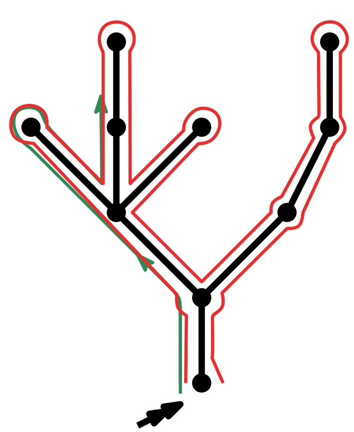
If T is a labeled tree, $(C_n(i),Z_n(i))$ = contour and label processes

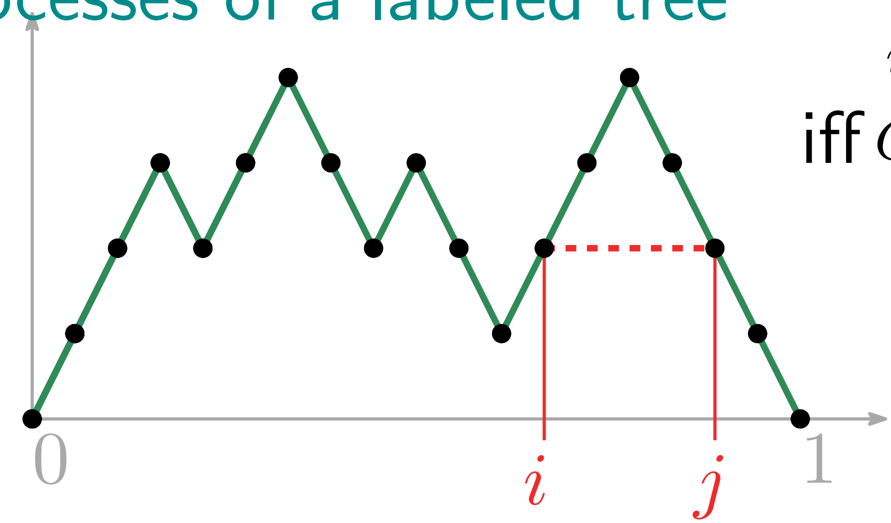
Convergence of labeled trees

Theorem : [Marckert '08]
For a sequence of simple random outer-triangular maps (M_n), the contour and label process of the associated labeled tree satisfie:

$$\Big((8n)^{-1/2}C_{\lfloor nt\rfloor},(1/2n)^{1/4}\tilde{Z}_{\lfloor nt\rfloor}\Big)_{0\le t\le 1}\ \xrightarrow[n\to\infty]{(d)}\ (e_t,Z_t)_{0\le t\le 1},$$

Contour and label processes of a labeled tree





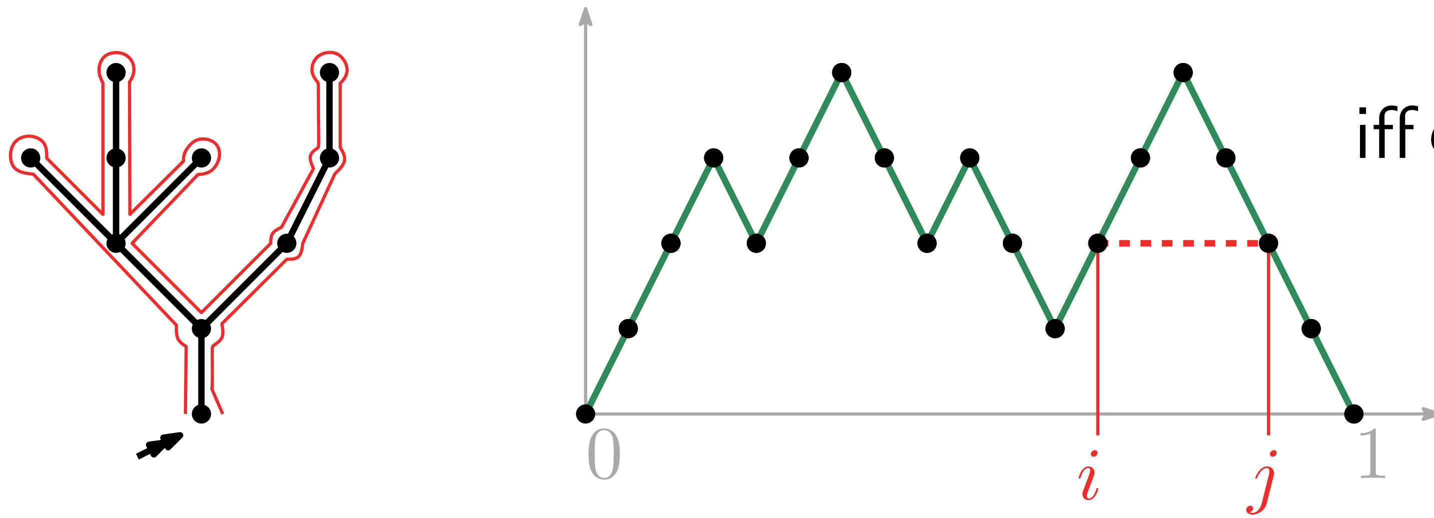
i and j = same vertex of T
iff $C_n(i)=C_n(j)=\min_{i\le k\le j} C_n(k)$

T ⟶ C_n^T (or C_n) = contour process

If T is a labeled tree, $(C_n(i),Z_n(i))$ = contour and label processes

Brownian snake $(e_t, Z_t)_{0 \le t \le 1}$

1st step : the Brownian tree [Aldous]

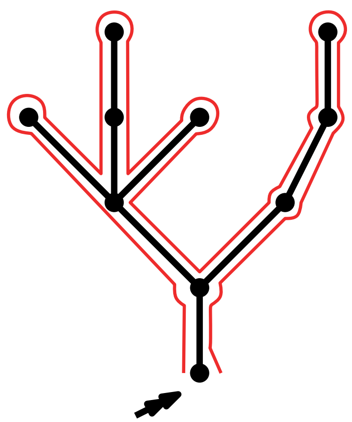


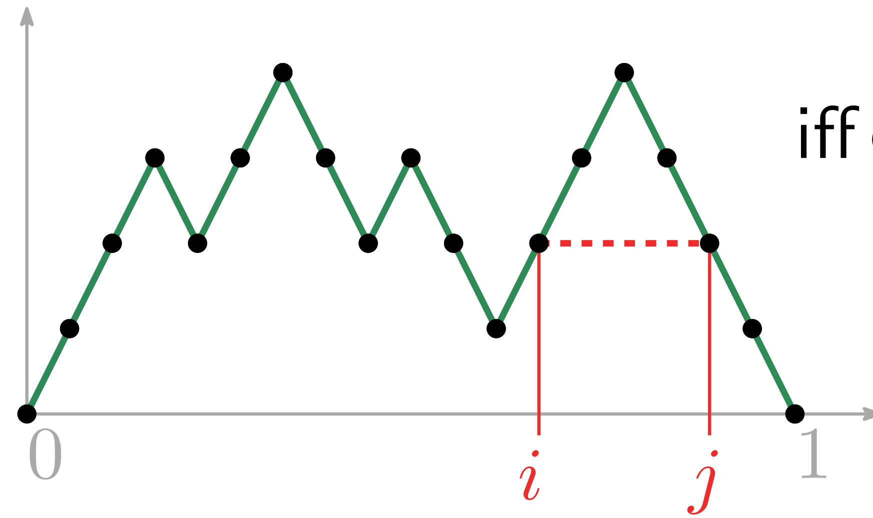
i and j = same vertex of T
iff $C_n(i) = C_n(j) = \min_{i \le k \le j} C_n(k)$

$T \longrightarrow C_n^T$ (or C_n) = contour process

Brownian snake $(e_t, Z_t)_{0\leq t\leq 1}$

1st step : the Brownian tree [Aldous]



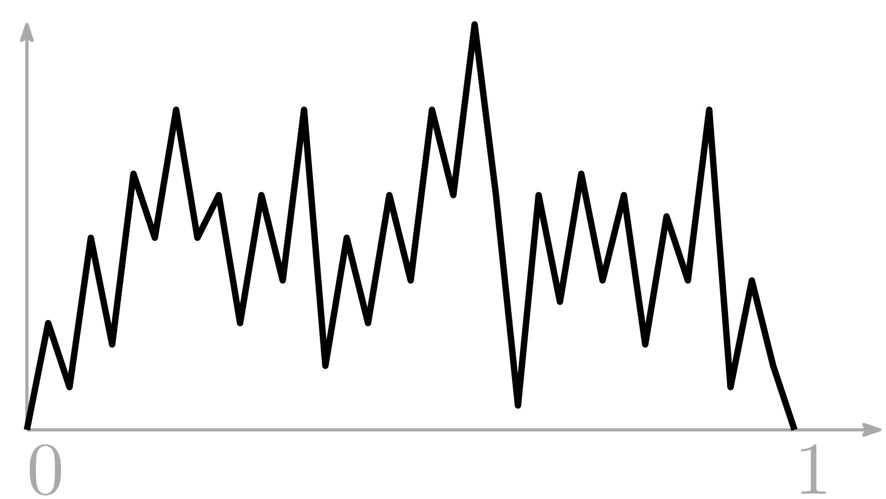


i and j = same vertex of T

iff $C_n(i) = C_n(j) = \min_{i\leq k\leq j} C_n(k)$

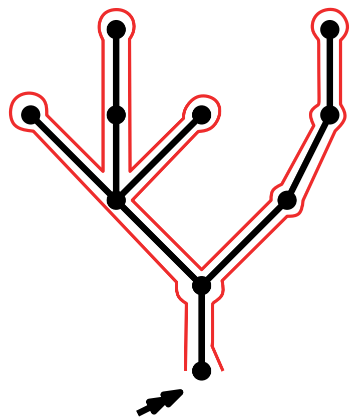
T $\longrightarrow$ C_n^T (or C_n) = contour process

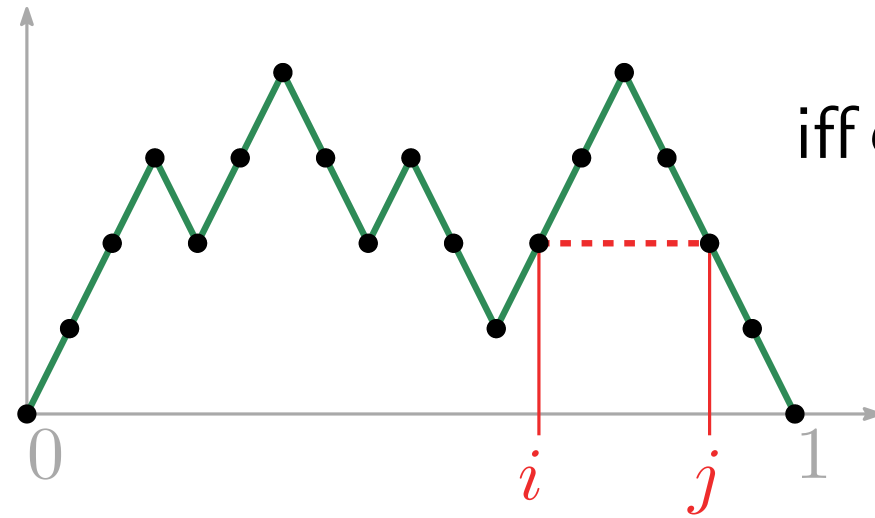
$(e_t)_{0\leq t\leq 1}$= Brownian excursion



Brownian snake $(e_t, Z_t)_{0\le t\le 1}$

1st step : the Brownian tree [Aldous]



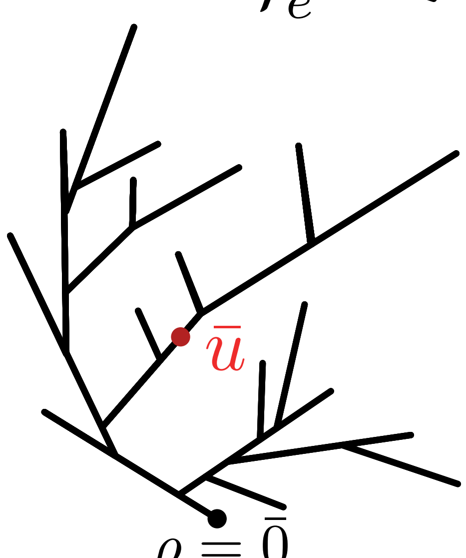


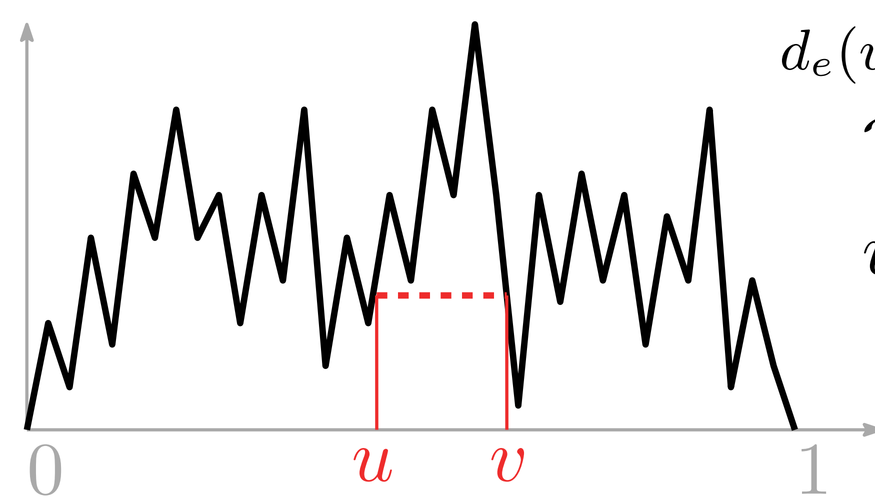
i and j = same vertex of T

iff $C_n(i) = C_n(j) = \min_{i\le k\le j} C_n(k)$

$T \longrightarrow C_n^T$ (or C_n) = contour process

$\mathcal{T}_e \longleftarrow (e_t)_{0\le t\le 1}$ = Brownian excursion





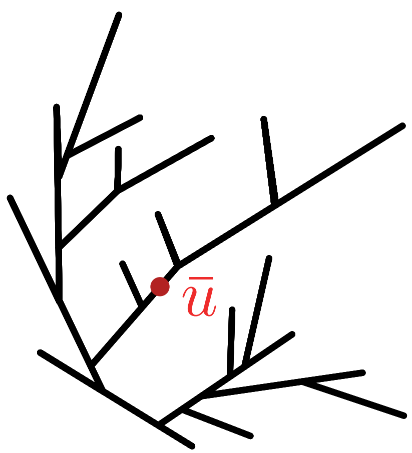
$d_e(u,v) = e_u + e_v - 2\min_{u\le s\le v} e_s$

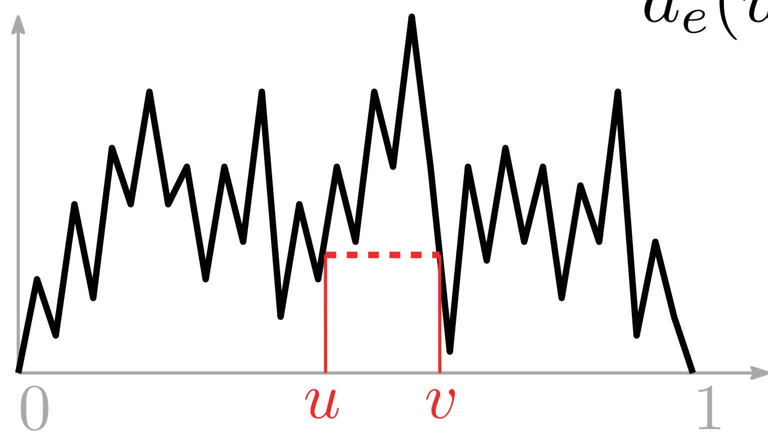
$\mathcal{T}_e = [0,1]/\sim_e$

$u \sim_e v$ iff $d_e(u,v) = 0$

Brownian snake $(e_t, Z_t)_{0\leq t\leq 1}$

1st step : the Brownian tree [Aldous]





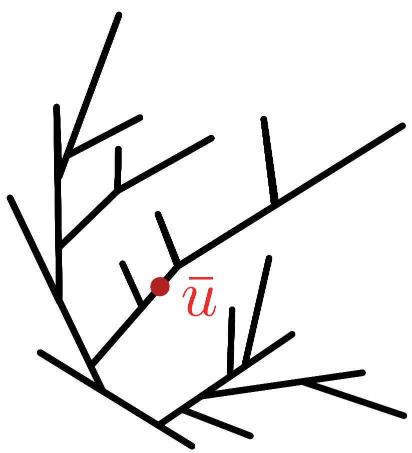
$d_e(u,v) = e_u + e_v - 2\min_{u\leq s\leq v} e_s$

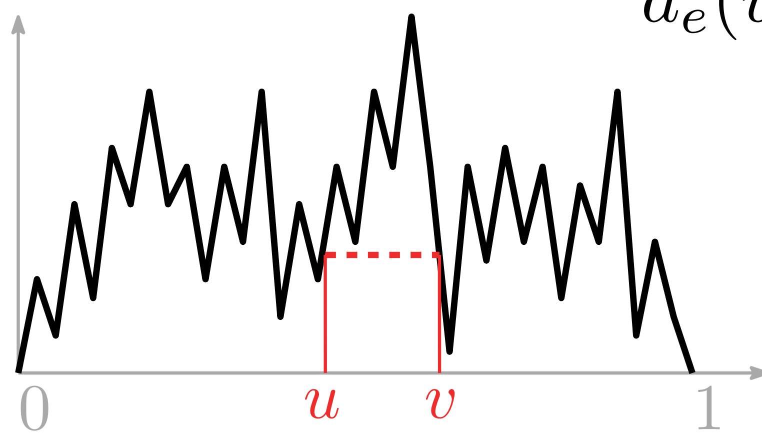
$\mathcal{T}_e = [0,1]/\sim_e$

$u \sim_e v$ iff $d_e(u,v) = 0$

Brownian snake $(e_t, Z_t)_{0\leq t\leq 1}$

1st step : the Brownian tree [Aldous]





$d_e(u,v) = e_u + e_v - 2\min_{u\leq s\leq v} e_s$

$\mathcal{T}_e = [0,1]/\sim_e$

$u \sim_e v$ iff $d_e(u,v) = 0$

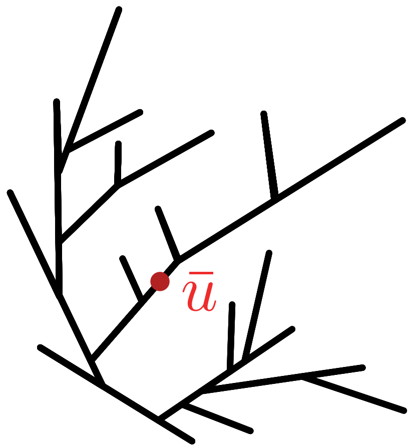
2nd step : Brownian labels

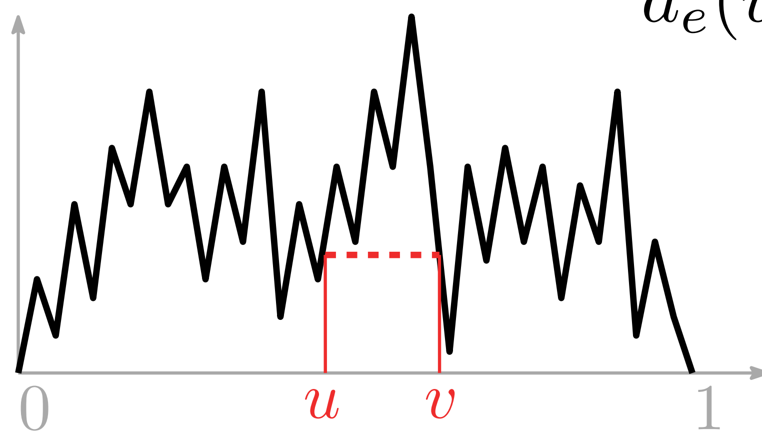
Conditional on $\mathcal{T}_e$, Z a centered Gaussian process with $Z_\rho = 0$ and $E[(Z_s - Z_t)^2] = d_e(s,t)$

$Z \sim$ Brownian motion on the tree

Brownian snake $(e_t, Z_t)_{0\leq t\leq 1}$

1st step : the Brownian tree [Aldous]





$d_e(u,v) = e_u + e_v - 2\min_{u\leq s\leq v} e_s$

$\mathcal{T}_e = [0,1]/\sim_e$

$u \sim_e v$ iff $d_e(u,v)=0$

2nd step : Brownian labels

Conditional on $\mathcal{T}_e$, Z a centered Gaussian process with $Z_\rho = 0$ and $E[(Z_s - Z_t)^2] = d_e(s,t)$

$Z \sim$ **Brownian motion on the tree**

Theorem :

$$\left((8n)^{-1/2} C_{\lfloor nt\rfloor}, (2n/1)^{-1/4}\tilde{Z}_{\lfloor nt\rfloor}\right)_{0\leq t\leq 1} \xrightarrow[n\to\infty]{(d)} (e_t, Z_t)_{0\leq t\leq 1},$$

Distances in simple outer-triangular maps

To do that :

- encode the maps by some trees.
- study the limits of trees,
- **interpret the distance in the maps by some function of the tree**.

S_n = outer-triangular simple map
$(C_{\lfloor nt \rfloor}, \tilde{Z}_{\lfloor nt \rfloor})$ = contour and label process of the associated tree
$Z_{\lfloor nt \rfloor}$ = distance **in the map** between vertex "$\lfloor nt \rfloor$" and the root.

Theorem :
S_n= random outer-triangular simple map, then for all $\varepsilon > 0$:

$$\mathbb{P}\left(\sup_{0 \le t \le 1} \left\{ \left| \tilde{Z}_{\lfloor nt \rfloor} - Z_{\lfloor nt \rfloor} \right| \right\} \ge \varepsilon n^{1/4} \right) \to 0.$$

i.e. the label process of the tree gives the distance to the root in the map.

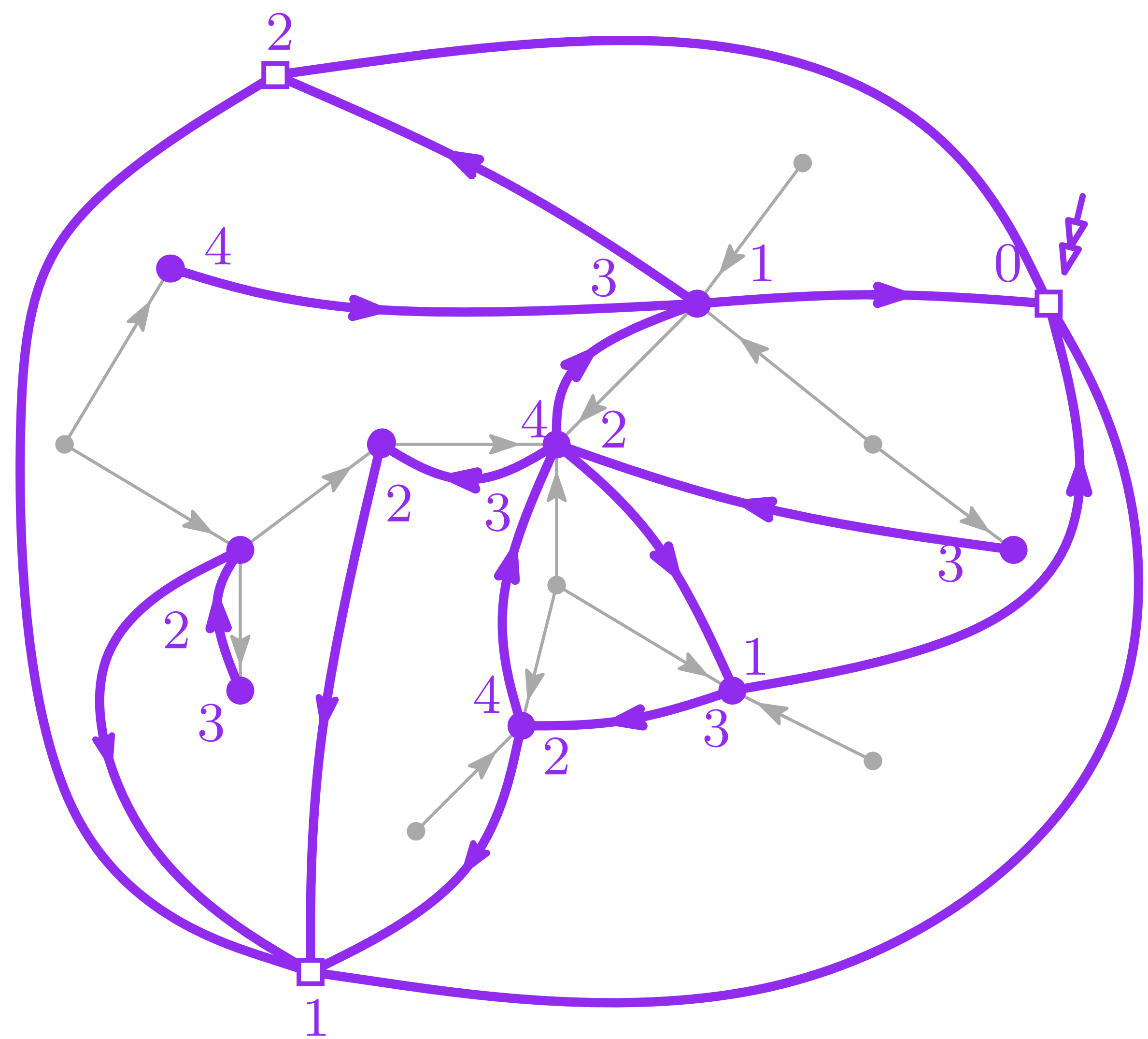
Distances in simple maps

Claim : $d_M(root, u) \leq$ Label of u

Distances in simple maps

Claim : $d_M(root, u) \leq$ Label of u

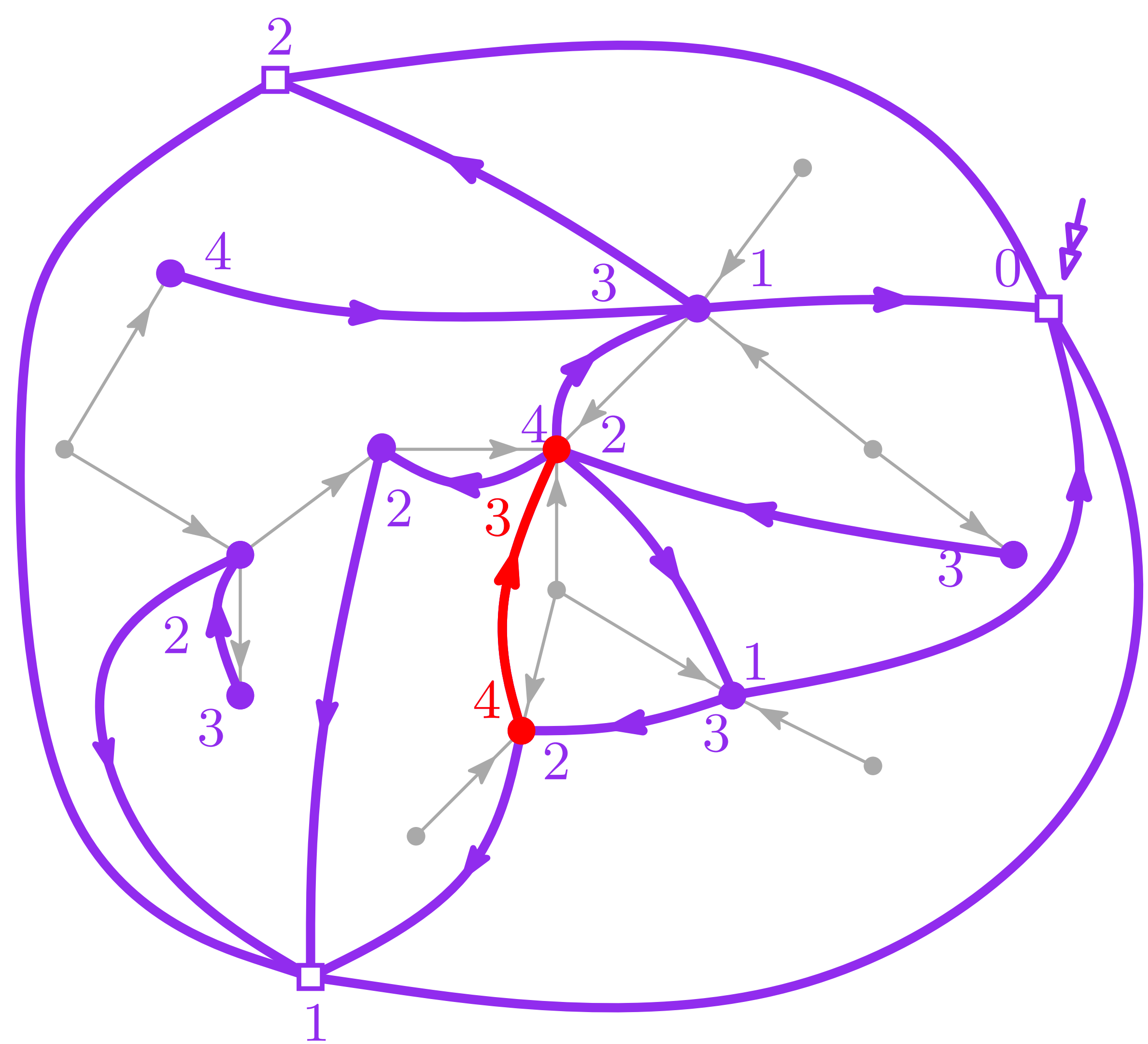
- Consider the **Left Most Path** from (u, v) to the root face.



Distances in simple maps

Claim : $d_M(root, u) \leq$ Label of u

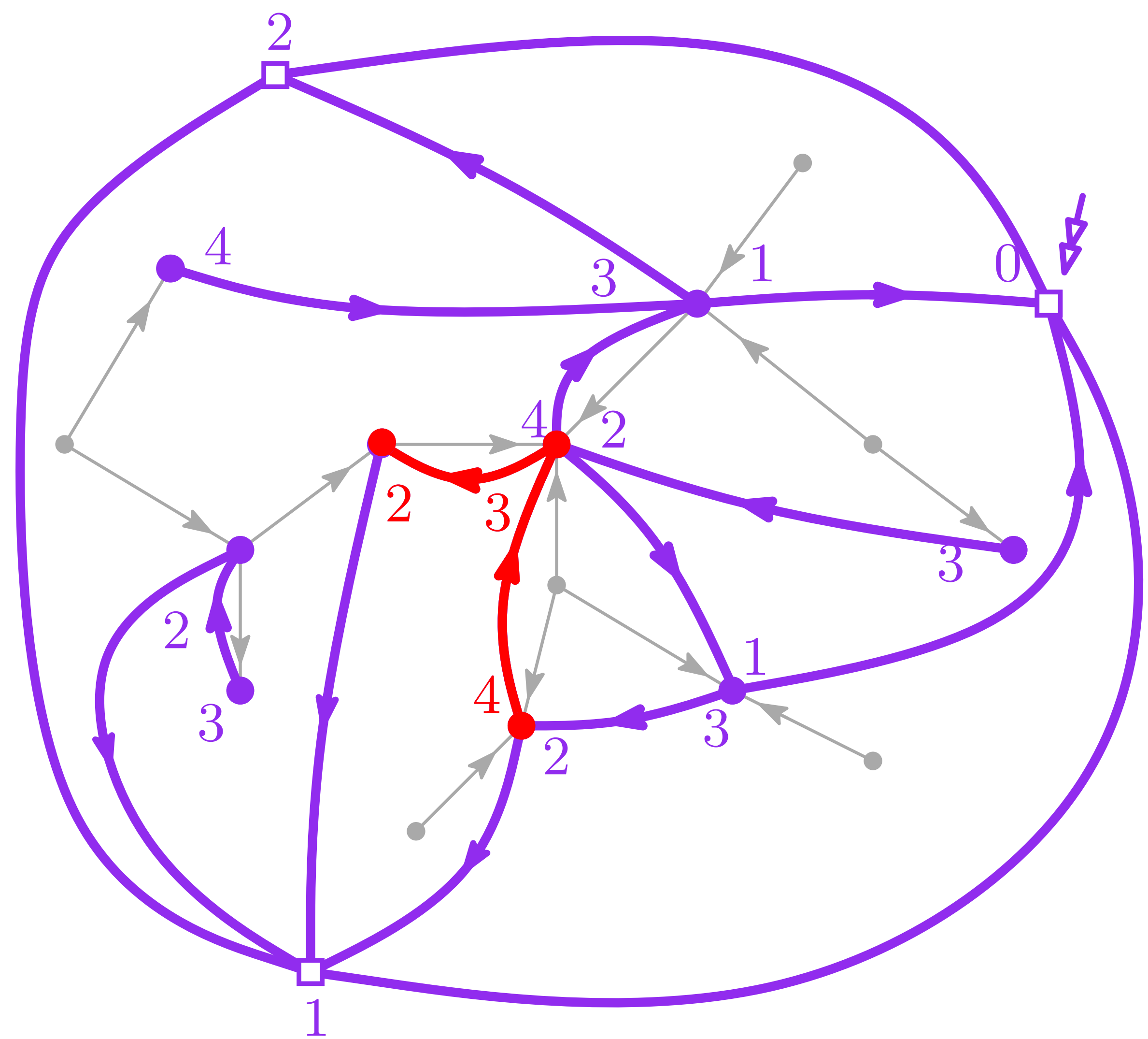
- Consider the **Left Most Path** from (u, v) to the root face.



Distances in simple maps

Claim : $d_M(root, u) \leq$ Label of u

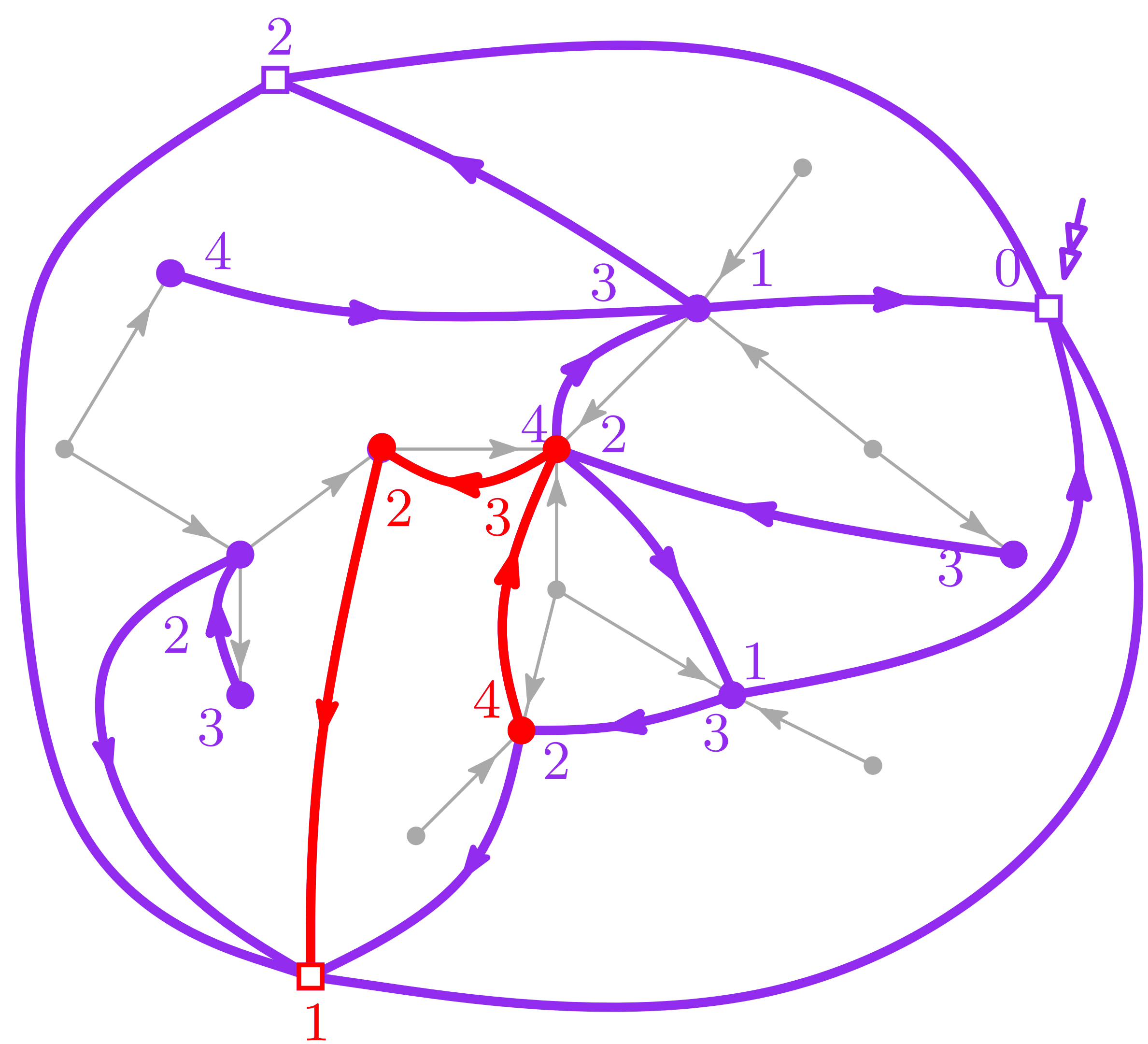
- Consider the **Left Most Path** from (u, v) to the root face.



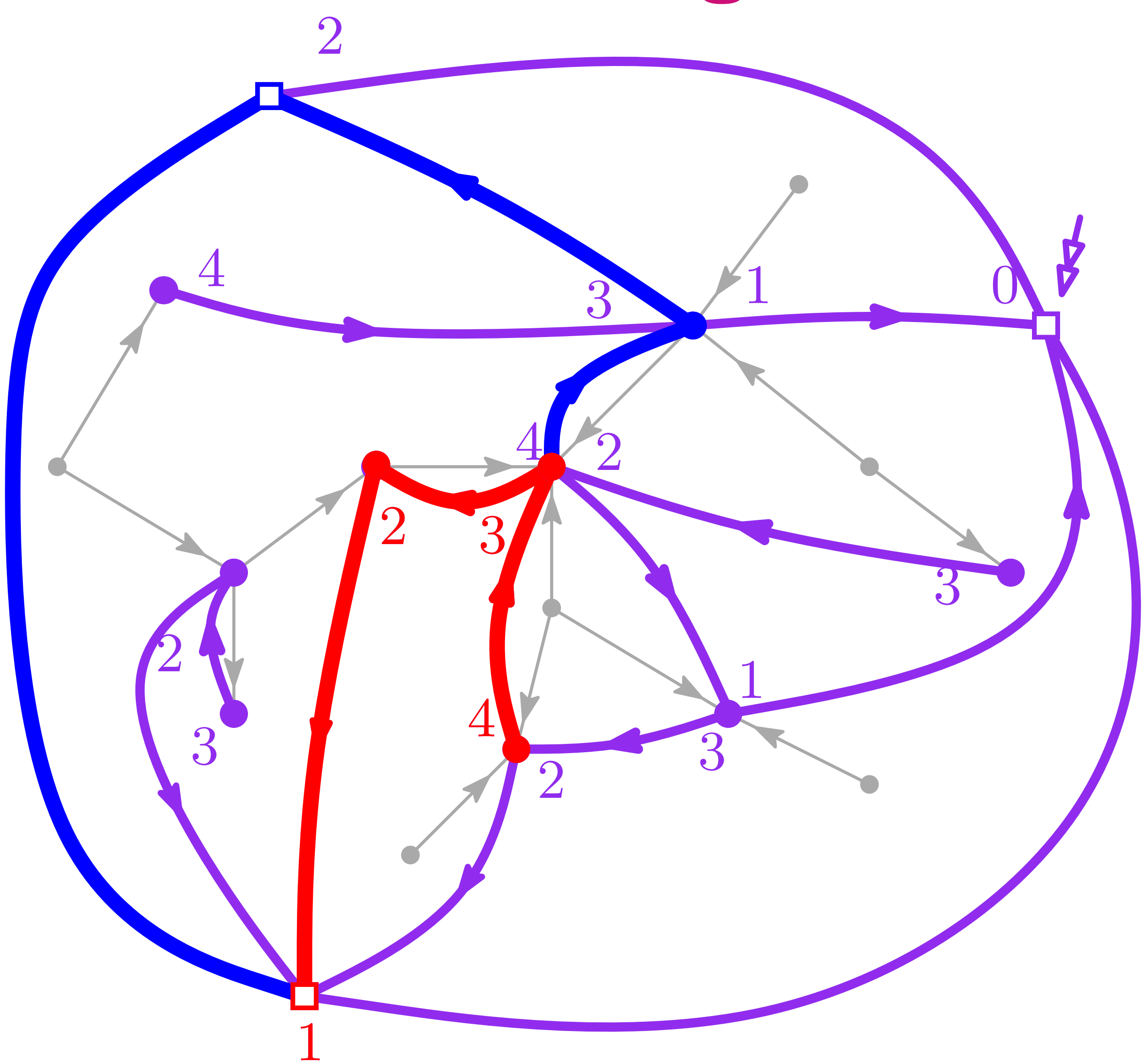
Distances in simple maps

Claim : $d_M(root, u) \leq$ Label of u

- Consider the **Left Most Path** from (u, v) to the root face.
- From the property of the closure, on the left of the LMP the labels decrease exactly by 1.
- The LMP is not self-intersecting: it reaches the outer-face



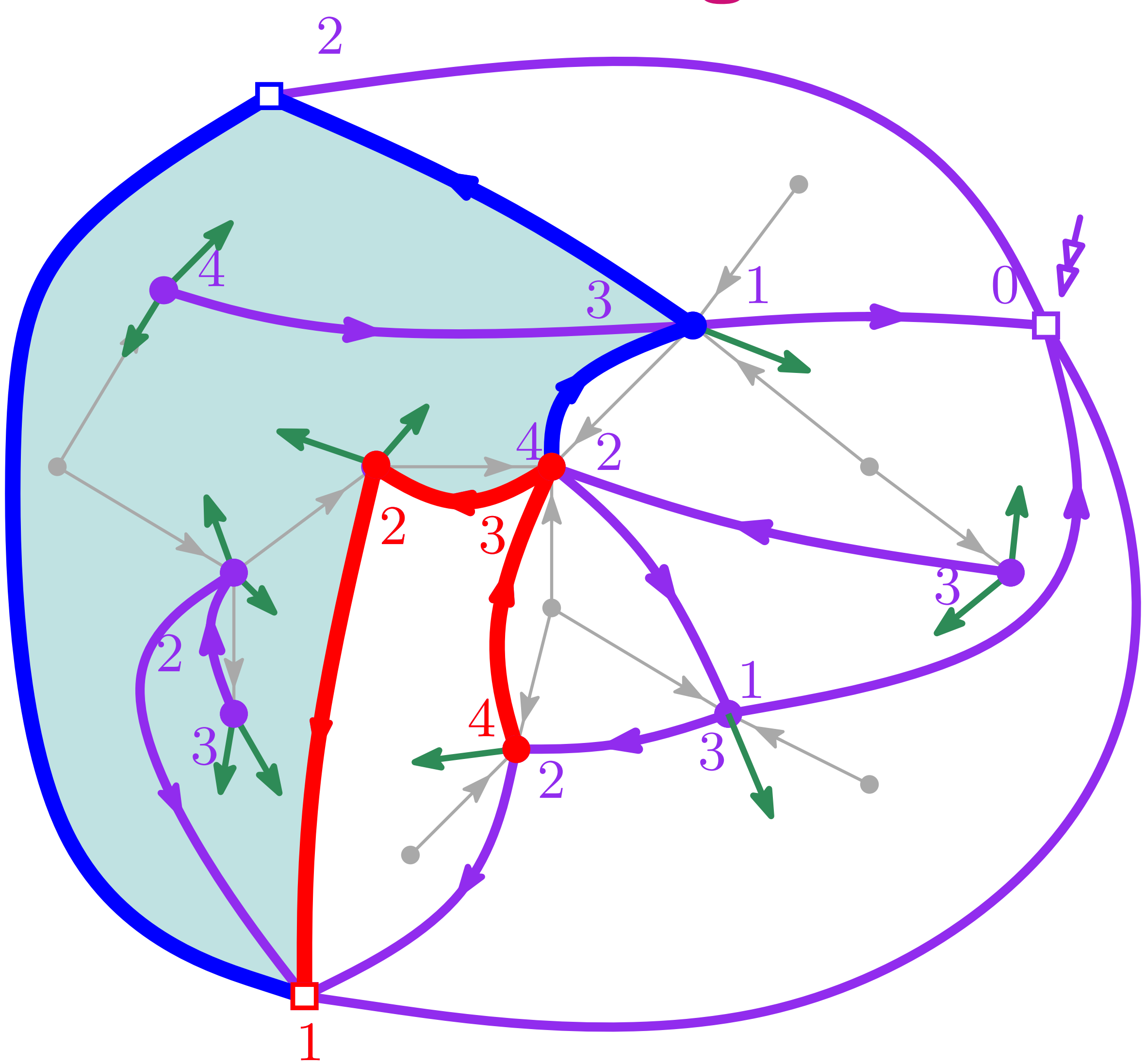
LMP are almost geodesic



Leftmost path

Another path: can it be shorter ?

LMP are almost geodesic

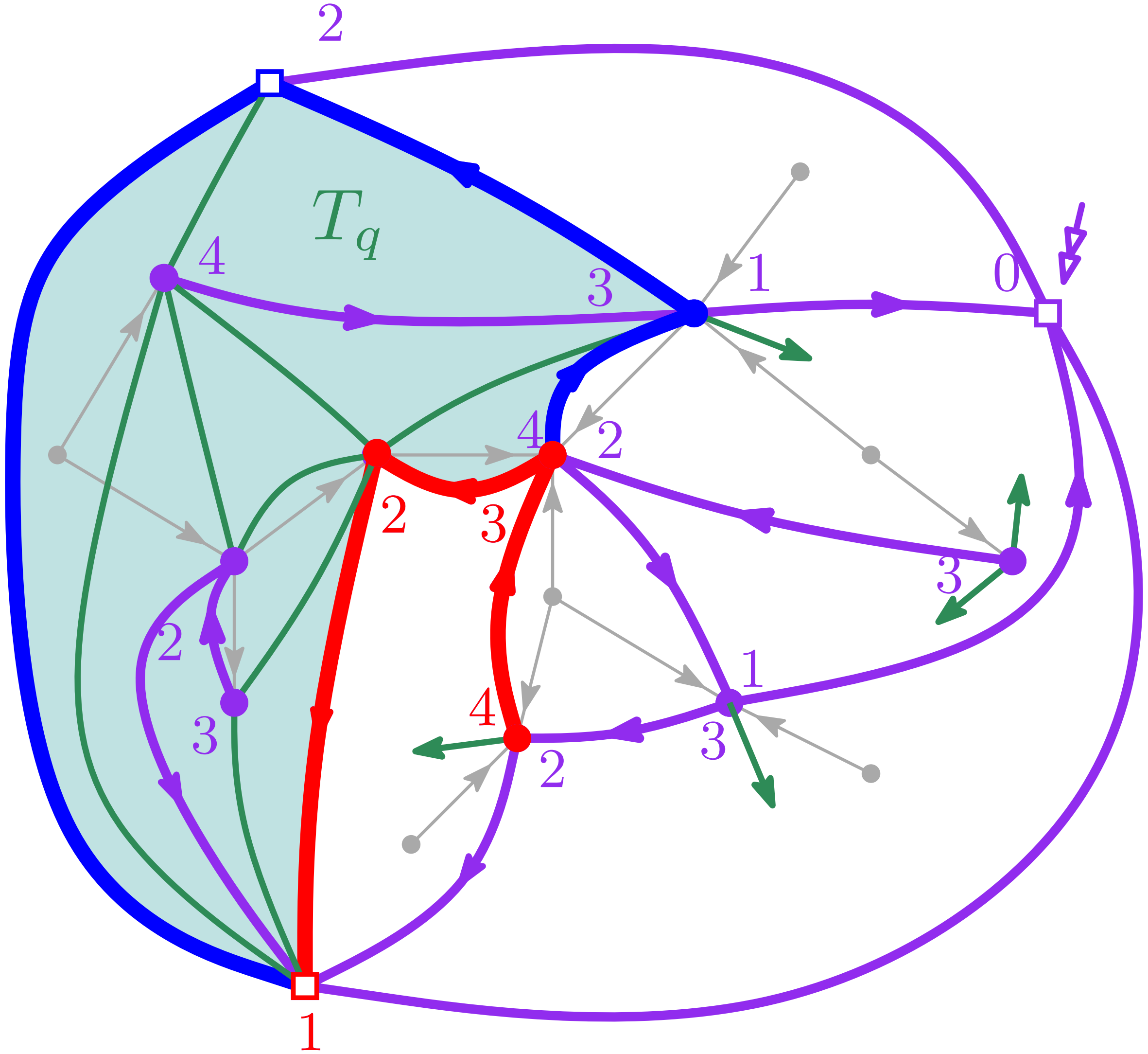


Consider the 3-orientation of the map with buds

Leftmost path

Another path: can it be shorter ?

LMP are almost geodesic



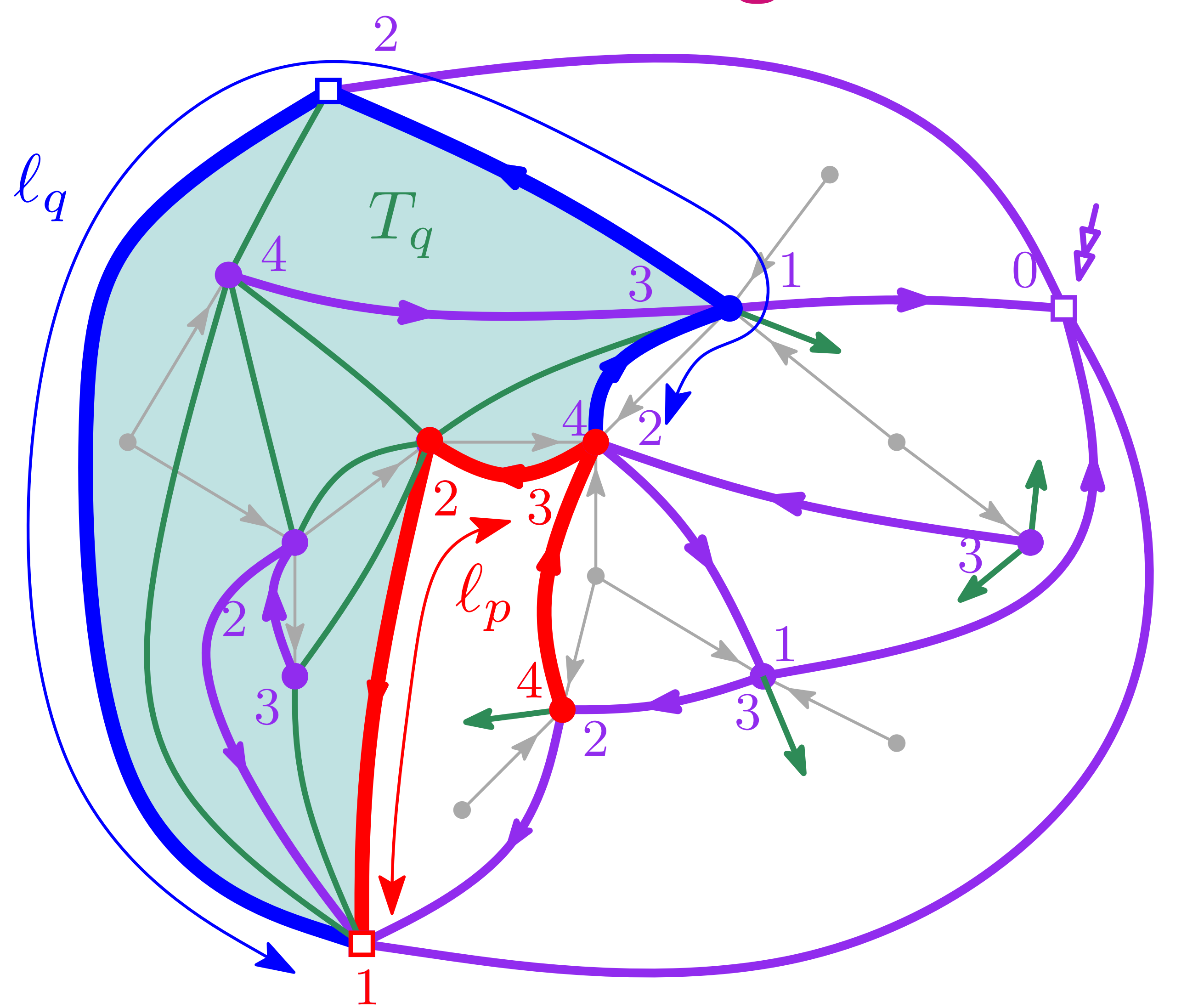
Consider the 3-orientation of the map with buds

Use the buds to triangulate the submap surrounded by the two paths.

Leftmost path

Another path: can it be shorter ?

LMP are almost geodesic



Leftmost path

Another path: can it be shorter ?

Consider the 3-orientation of the map with buds

Use the buds to triangulate the submap surrounded by the two paths.

Euler Formula :

$$|E(T_q)| = 3|V(T_q)| - 3 - (\ell_p + \ell_q)$$

3-orientation + LMP :

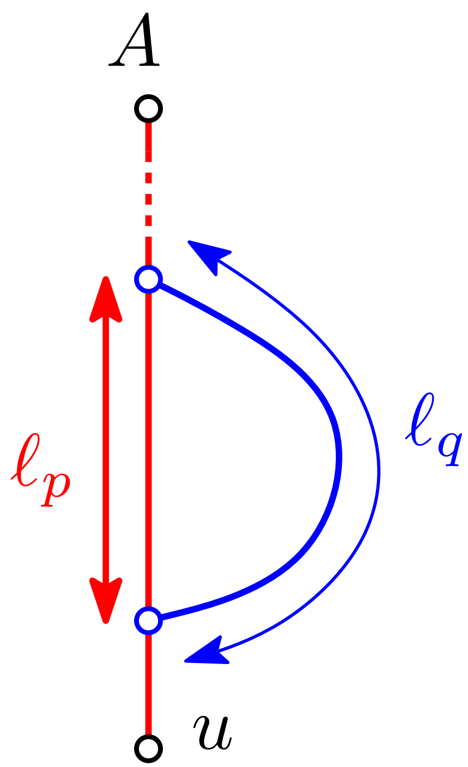
$$|E(T_q)| \geq 3|V(T_q)| - 2\ell_q - 2$$

$$\Longrightarrow \ell_q \geq \ell_p + 1$$

LMP are almost geodesic

Leftmost path

Another path: can it be shorter ?

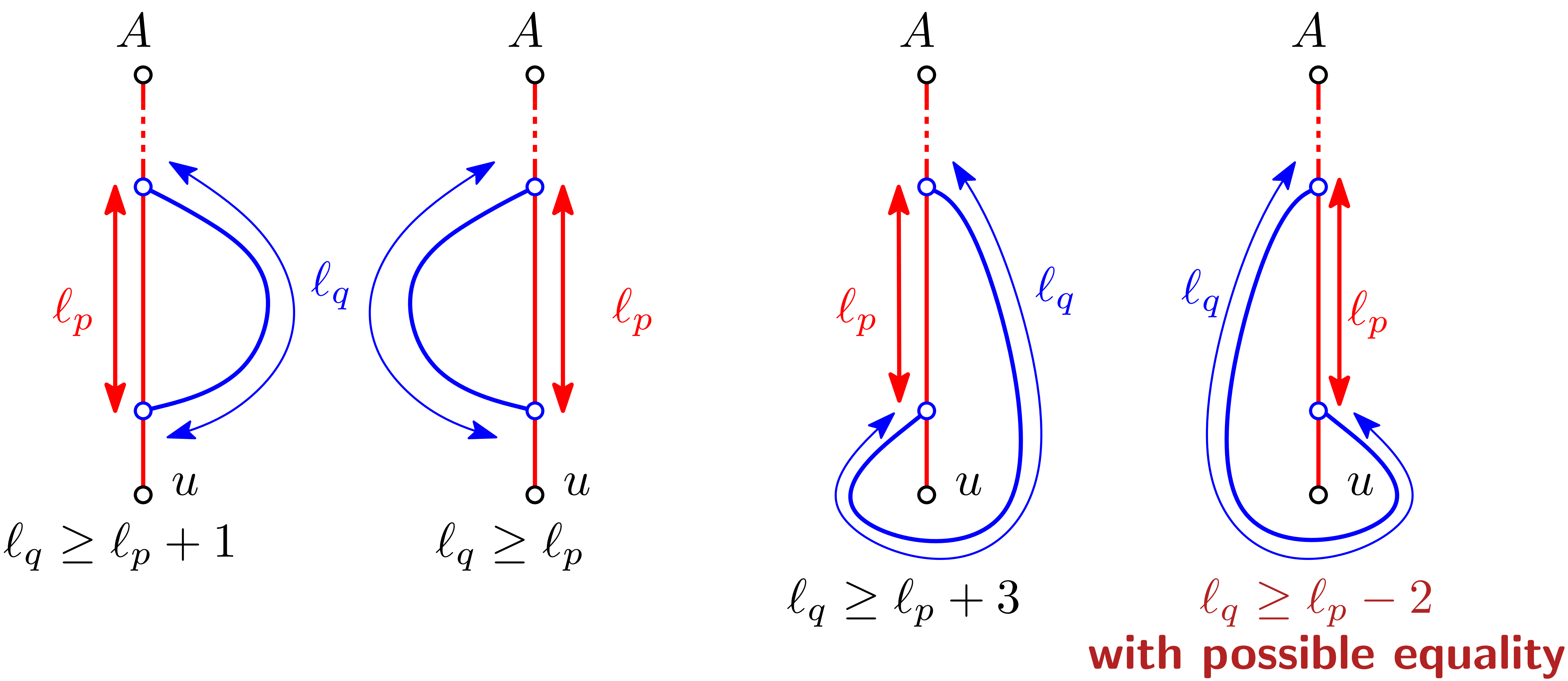


$$\ell_q \geq \ell_p + 1$$

LMP are almost geodesic

Leftmost path

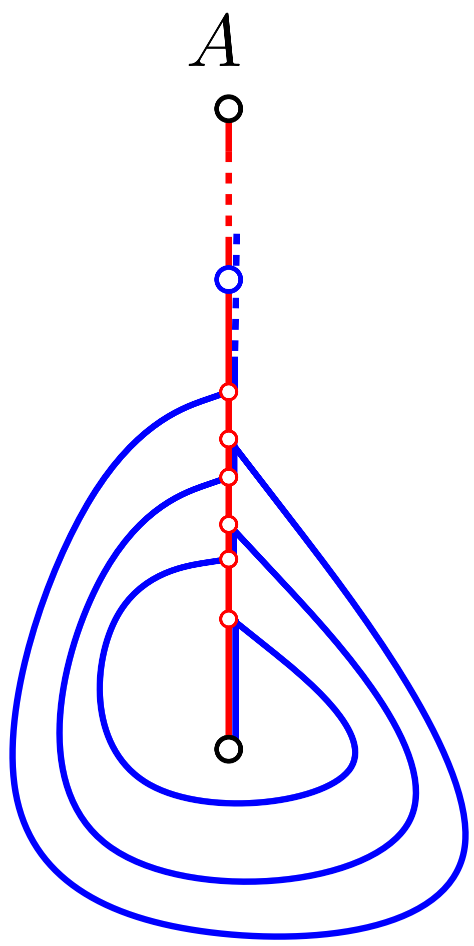
Another path: can it be shorter ? YES



LMP are almost geodesic

Leftmost path

Another path: can it be shorter ? YES ... but not too often



Bad configuration =
too many **windings** around the LMP

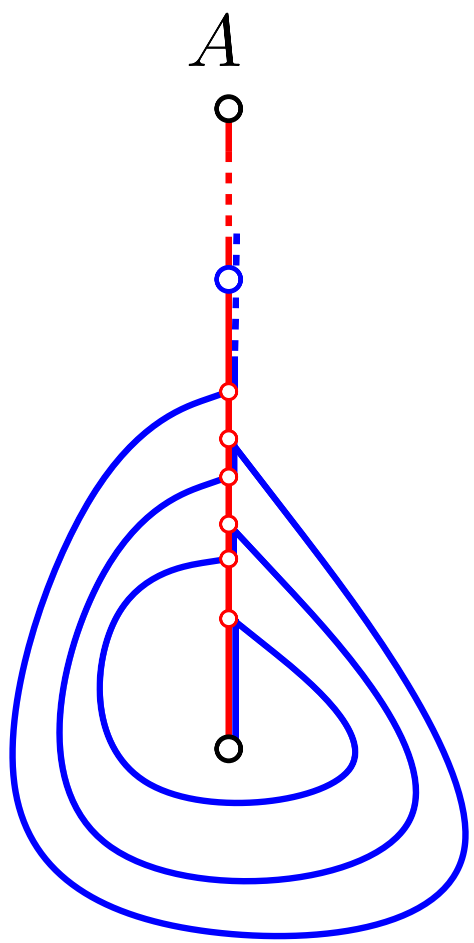
But w.h.p a winding cannot be too short.

$\Longrightarrow$ w.h.p the number of windings is $o(n^{1/4})$.

LMP are almost geodesic

Leftmost path

Another path: can it be shorter ? YES ... but not too often



Bad configuration =
too many **windings** around the LMP

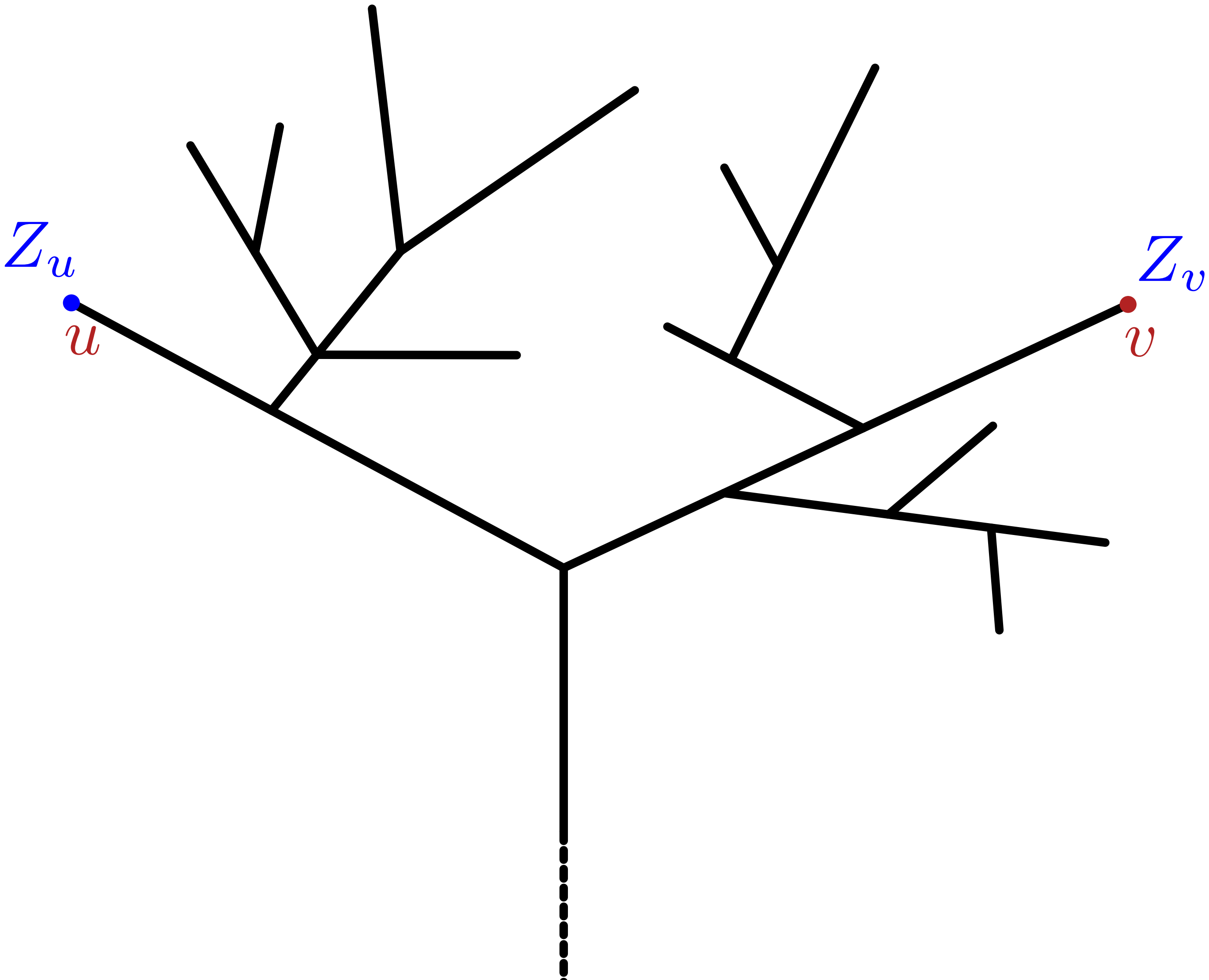
But w.h.p a winding cannot be too short.

$\Longrightarrow$ w.h.p the number of windings is $o(n^{1/4})$.

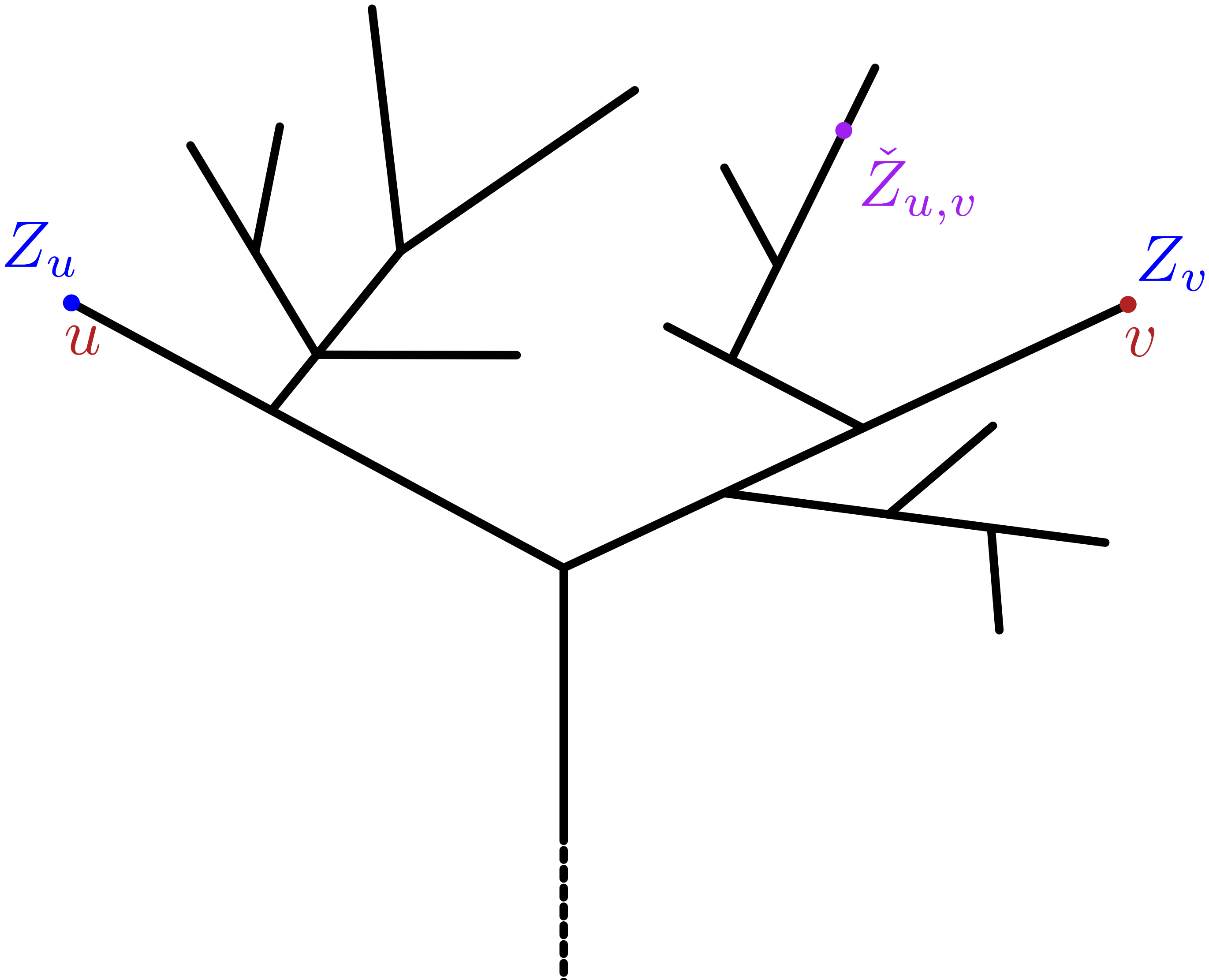
Proposition: [Addario-Berry, A. '13]
For $\varepsilon > 0$, let $A_{n,\varepsilon}$ be the event that there exists $u \in M_n$ such that
Label of $u \geq d_{M_n}(u, root) + \varepsilon n^{1/4}$.
Then under the uniform law on $\mathcal{M}_n$, for all $\varepsilon > 0$:

$$\mathbb{P}\left(A_{n,\varepsilon}\right) \to 0.$$

Distances are tight

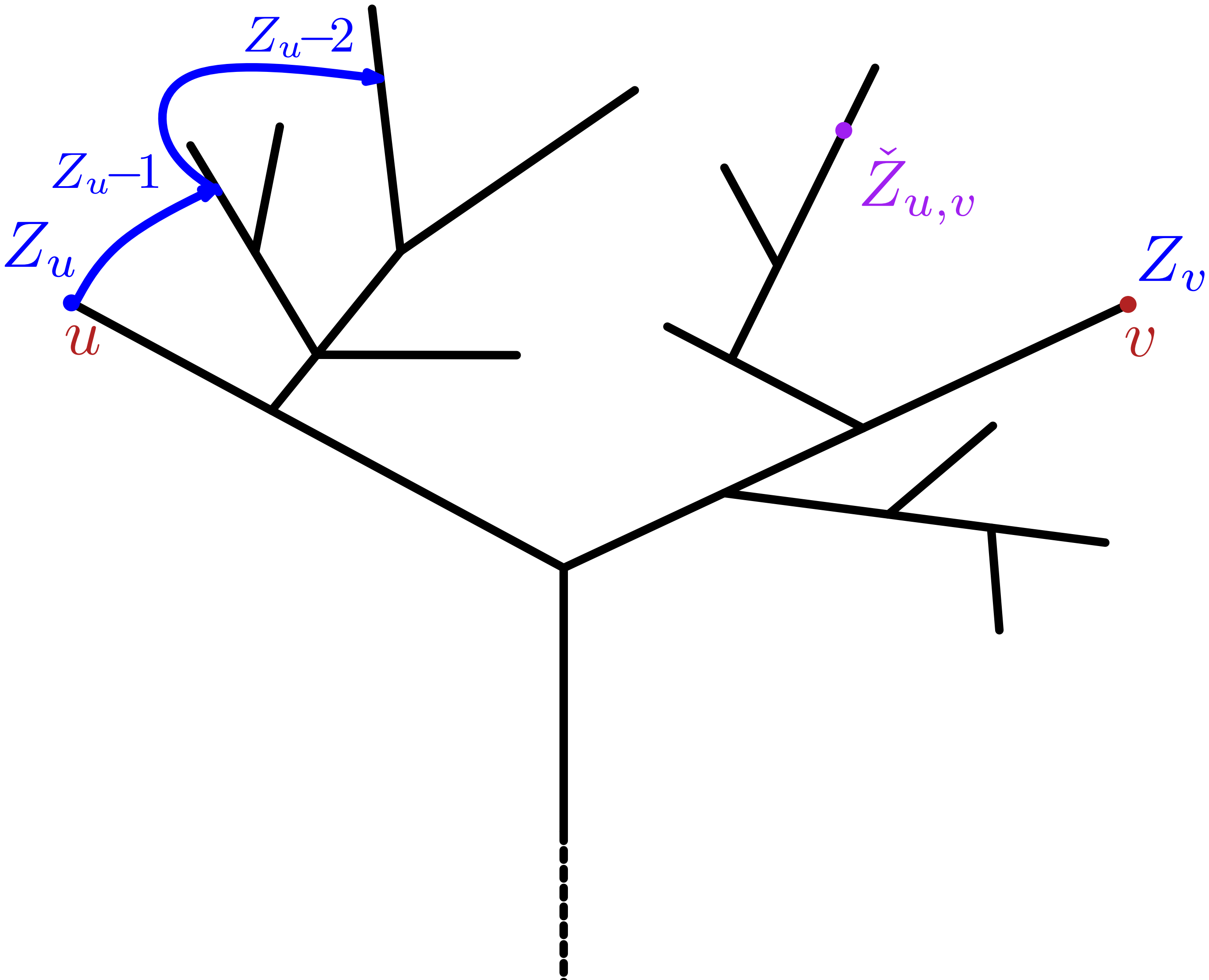


Distances are tight



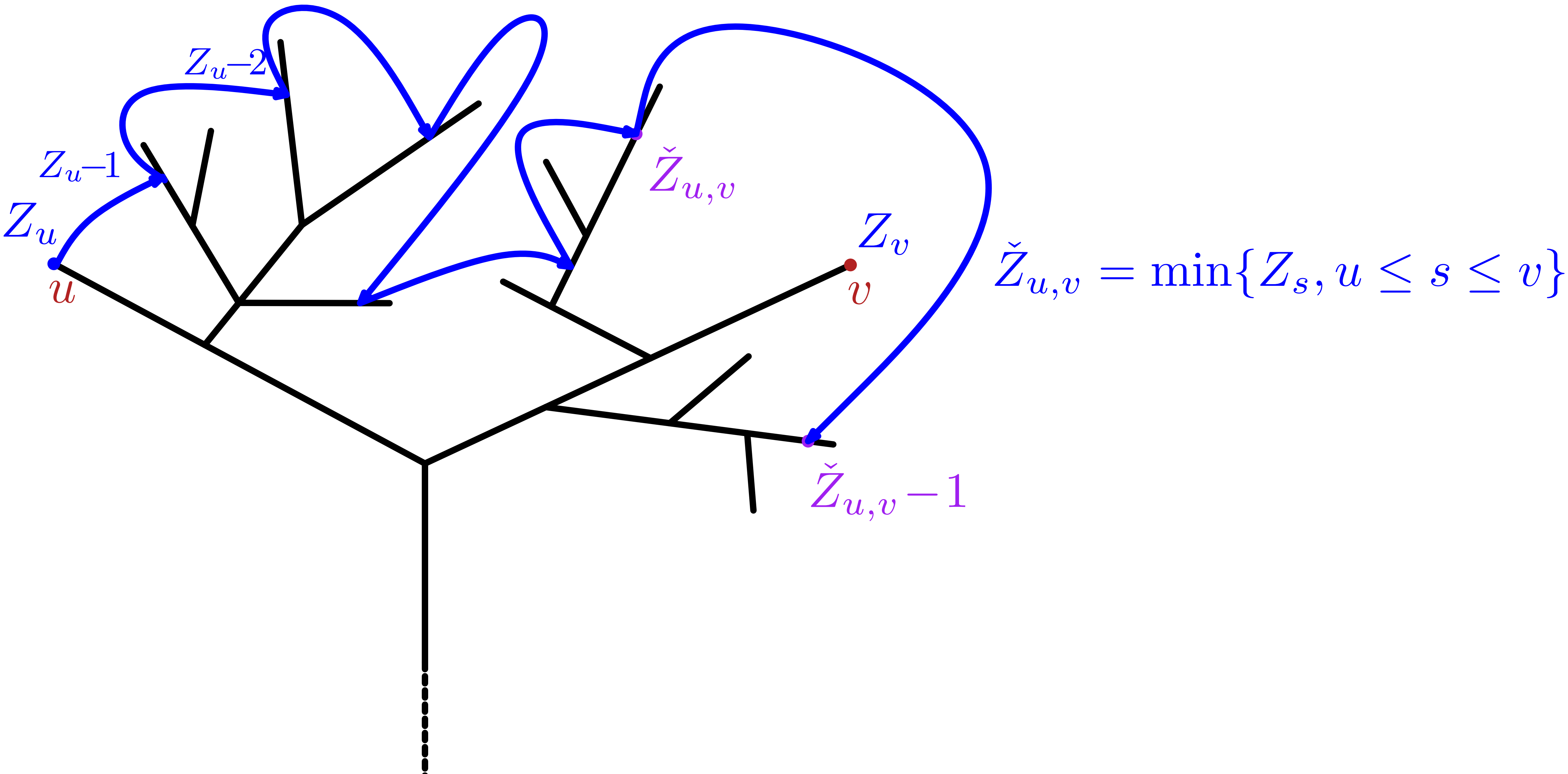
$$\check{Z}_{u,v} = \min\{Z_s, u \leq s \leq v\}$$

Distances are tight

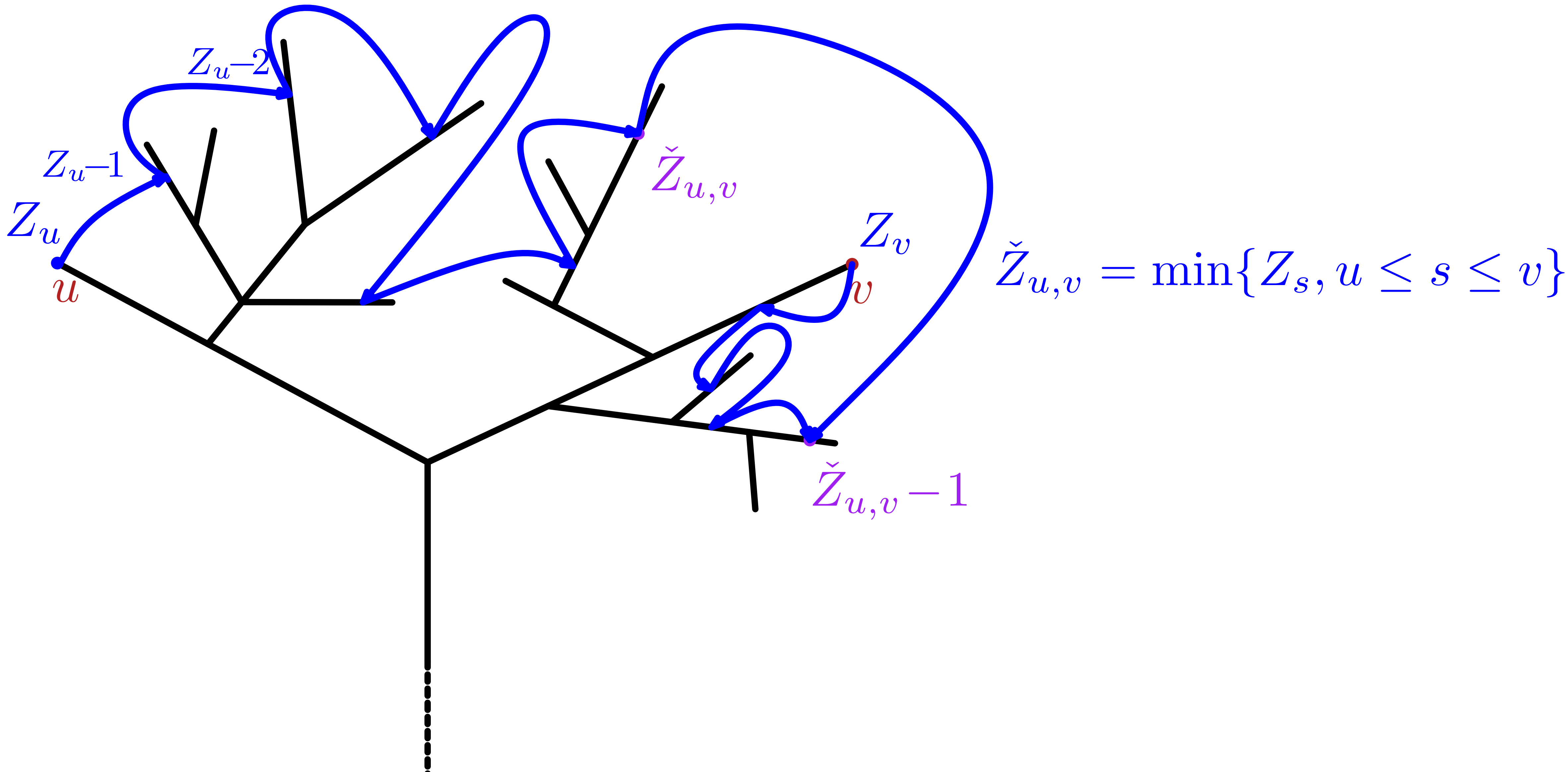


$$\check{Z}_{u,v} = \min\{Z_s, u \le s \le v\}$$

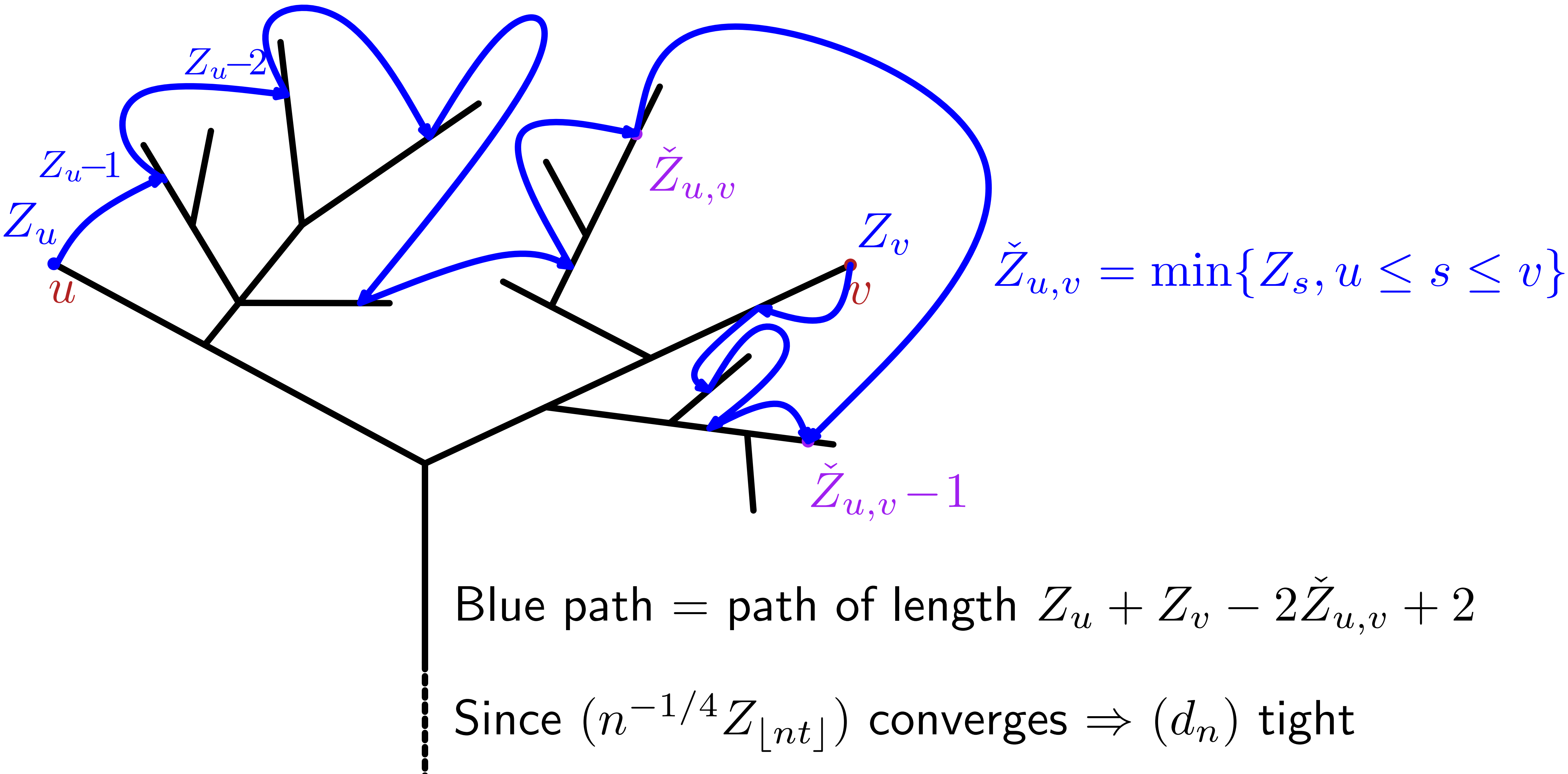
Distances are tight



Distances are tight



Distances are tight



Blue path = path of length $Z_u + Z_v - 2\check{Z}_{u,v} + 2$

Since $(n^{-1/4} Z_{\lfloor nt \rfloor})$ converges $\Rightarrow$ (d_n) tight

The result for the last time

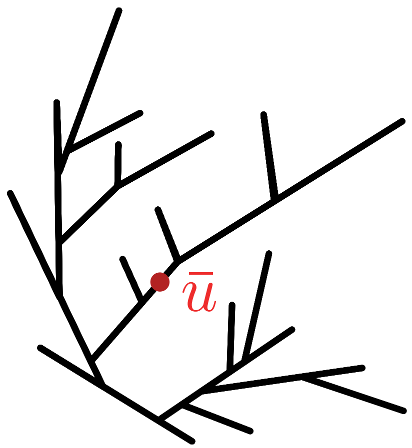
Theorem : [A., Bernardi, Collet, Fusy]
$\mathcal{S}_n = \{$ simple maps with n edges $\}$
$S_n =$ uniform random element of $\mathcal{S}_n$. Then:

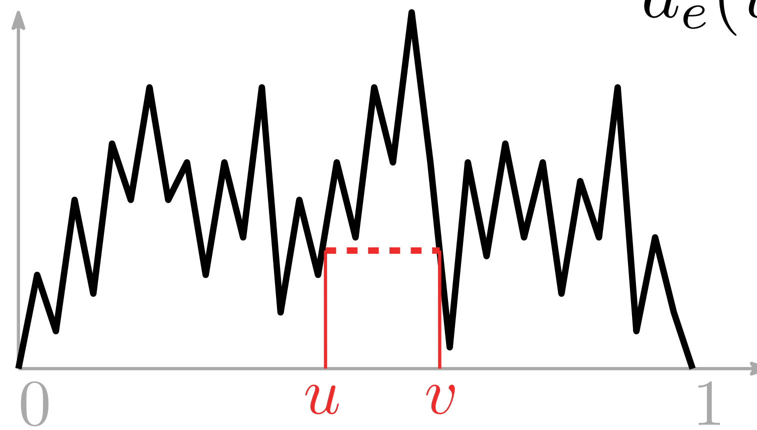
$$\left(S_n, \left(\frac{1}{2n}\right)^{1/4} d_{S_n}\right) \xrightarrow{(d)} (M, D^\star),$$

for the distance of Gromov-Hausdorff on the isometry classes of compact metric spaces.

The Brownian Map ??

The Brownian map





$d_e(u,v) = e_u + e_v - 2\min_{u \le s \le v} e_s$

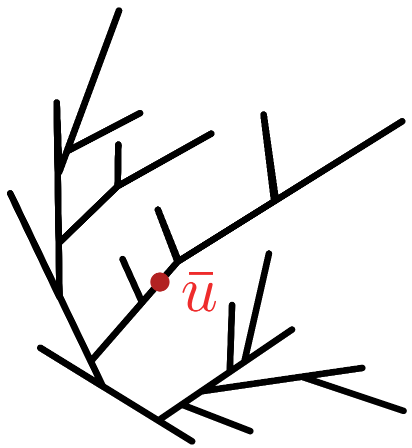
$\mathcal{T}_e = [0,1]/\sim_e$

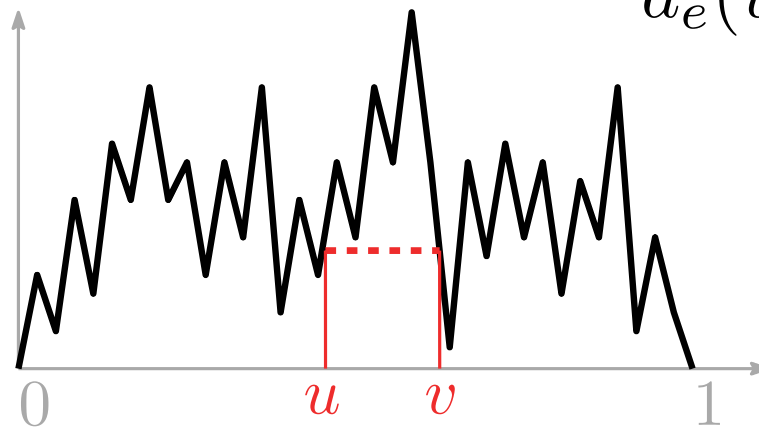
$u \sim_e v$ iff $d_e(u,v) = 0$

Conditional on $\mathcal{T}_e$, Z a centered Gaussian process with $Z_\rho = 0$ and $E[(Z_s - Z_t)^2] = d_e(s,t)$

$Z \sim$ Brownian motion on the tree

The Brownian map





$d_e(u,v) = e_u + e_v - 2\min_{u\le s\le v} e_s$

$\mathcal{T}_e = [0,1]/\sim_e$

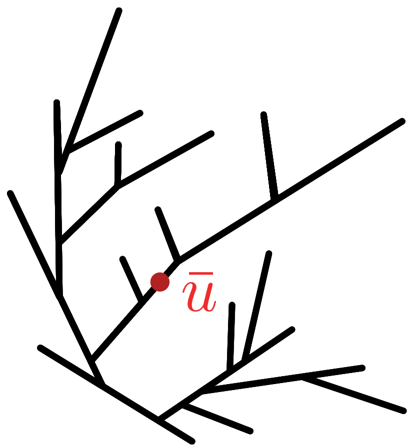
$u \sim_e v$ iff $d_e(u,v) = 0$

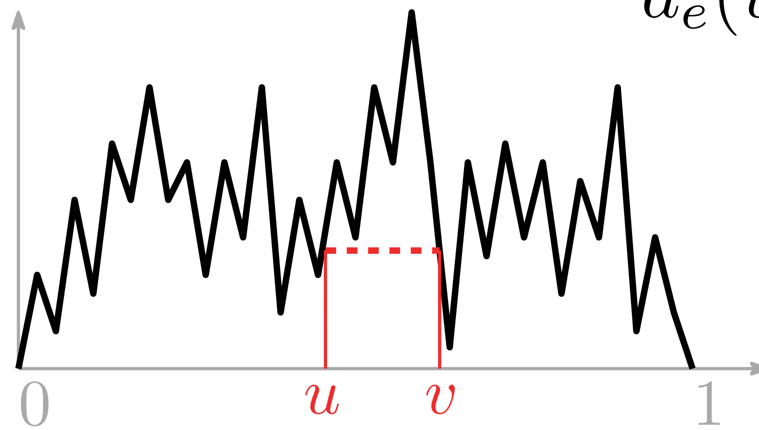
Conditional on $\mathcal{T}_e$, Z a centered Gaussian process with $Z_\rho = 0$ and $E[(Z_s - Z_t)^2] = d_e(s,t)$

$Z \sim$ Brownian motion on the tree

$$D^\circ(s,t) = Z_s + Z_t - 2\max\Big(\inf_{s\le u\le t} Z_u, \inf_{t\le u\le s} Z_u\Big), \quad s,t \in [0,1].$$

The Brownian map





$$d_e(u,v) = e_u + e_v - 2\min_{u \le s \le v} e_s$$

$$\mathcal{T}_e = [0,1]/\sim_e$$

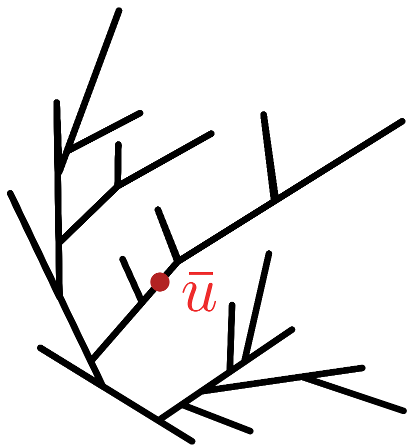
$u \sim_e v$ iff $d_e(u,v) = 0$

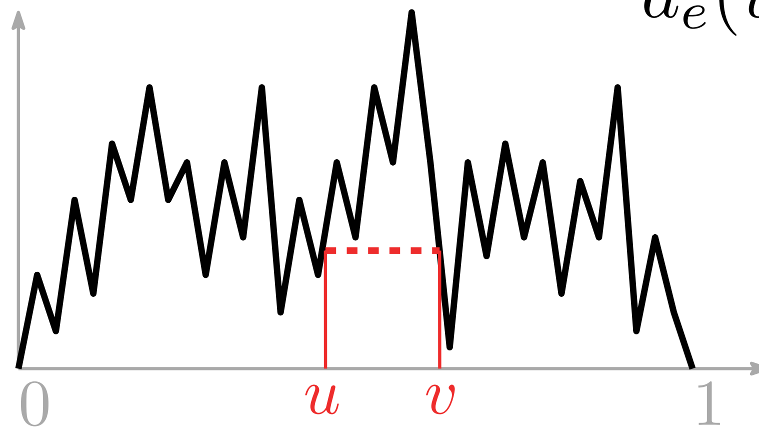
Conditional on $\mathcal{T}_e$, Z a centered Gaussian process with $Z_\rho = 0$ and $E[(Z_s - Z_t)^2] = d_e(s,t)$ **$Z \sim$ Brownian motion on the tree**

$$D^\circ(s,t) = Z_s + Z_t - 2\max\left(\inf_{s \le u \le t} Z_u, \inf_{t \le u \le s} Z_u\right), \quad s,t \in [0,1].$$

$$D^*(a,b) = \inf\left\{\sum_{i=1}^{k-1} D^\circ(a_i, a_{i+1}) : k \ge 1, a = a_1, a_2, \ldots, a_{k-1}, a_k = b\right\},$$

The Brownian map





$d_e(u,v) = e_u + e_v - 2\min_{u \le s \le v} e_s$

$\mathcal{T}_e = [0,1]/\sim_e$

$u \sim_e v$ iff $d_e(u,v) = 0$

Conditional on $\mathcal{T}_e$, Z a centered Gaussian process with $Z_\rho = 0$ and $E[(Z_s - Z_t)^2] = d_e(s,t)$ $Z \sim$ **Brownian motion on the tree**

$$D^\circ(s,t) = Z_s + Z_t - 2\max\left(\inf_{s \le u \le t} Z_u, \inf_{t \le u \le s} Z_u\right), \quad s,t \in [0,1].$$

$$D^*(a,b) = \inf\left\{\sum_{i=1}^{k-1} D^\circ(a_i, a_{i+1}) : k \ge 1, a = a_1, a_2, \ldots, a_{k-1}, a_k = b\right\},$$

Then $M = (\mathcal{T}_e/\sim_{D^*}, D^*)$ is the **Brownian map**.

A word of conclusion

Nice to see that the idea of LMP introduced for simple triangulations also work for simple maps.

Natural further step: try to adapt the techniques for all the bijections involving blossoming trees. In particular in the unified setting of [Bernardi, Fusy '10] and [A., Poulalhon '14]

A word of conclusion

Nice to see that the idea of LMP introduced for simple triangulations also work for simple maps.

Natural further step: try to adapt the techniques for all the bijections involving blossoming trees. In particular in the unified setting of [Bernardi, Fusy '10] and [A., Poulalhon '14]

Question: Can we make it work for 3-connected planar maps ?

A word of conclusion

Nice to see that the idea of LMP introduced for simple triangulations also work for simple maps.

Natural further step: try to adapt the techniques for all the bijections involving blossoming trees. In particular in the unified setting of [Bernardi, Fusy '10] and [A., Poulalhon '14]

Question: Can we make it work for 3-connected planar maps ?

By Whitney's theorem, a 3-connected planar **graph** has a unique embedding as a planar **map**.

$\Rightarrow$ would permit to get results about 3-connected planar graphs (and then about 2-connected and connected planar graphs).

A word of conclusion

Nice to see that the idea of LMP introduced for simple triangulations also work for simple maps.

Natural further step: try to adapt the techniques for all the bijections involving blossoming tr in the unified setting of [Bernardi, Fusy '10]

Thank you !

Question: Can w cted planar maps ?

By Whitney's theore ar **graph** has a unique embedding as a planar

⇒ would permit to get results about 3-connected planar graphs (and then about 2-connected and connected planar graphs).