Course IMSc Chennai, India

January-March 2017

Enumerative and algebraic combinatorics, a bijective approach: commutations and heaps of pieces (with interactions in physics, mathematics and computer science)

Monday and Thursday 14h-15h30

www.xavierviennot.org/coursIMSc2017

IMSc

January-March 2017

Xavier Viennot

CNRS, LaBRI, Bordeaux

www.xavierviennot.org

Chapter 3
Heaps and Paths,
Flows and Rearrangements monoids
(2)

IMSc, Chennai

30 January 2017

flow monoid (on X) $F(X)$

- X set
- $P = A = \{(i,j)\}_{\substack{i \in X \\ j \in X}}$
  basic pieces, alphabet
- $\mathcal{C}$ dependency relation: (or concurrency)
  $(i,j) \,\mathcal{C}\, (i',j') \iff i = i'$

C commutations

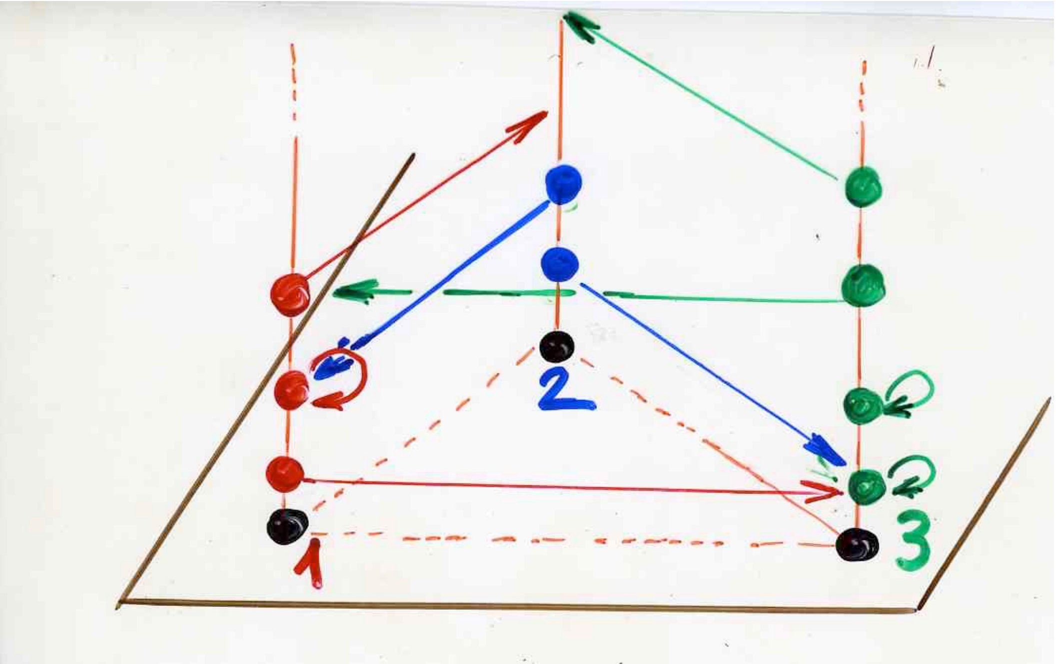
$(i,j)(i',j') = (i',j')(i,j)$ iff $i \neq i'$

$A = X \times X$ $\quad \begin{pmatrix} i \\ j \end{pmatrix}$

$[w]$ equivalence class of biwords for C

$$w = \begin{pmatrix} 1 & 3 & 2 & 3 & 1 & 3 & 2 & 1 & 3 \\ 3 & 3 & 3 & 3 & 1 & 1 & 1 & 2 & 2 \end{pmatrix}$$

$X = \{1, 2, 3\}$ flow



total order on X

$$w = \begin{pmatrix} 1 & 3 & 2 & 3 & 1 & 3 & 2 & 1 & 3 \\ 3 & 3 & 3 & 3 & 1 & 1 & 1 & 2 & 2 \end{pmatrix}$$

$$w \equiv_{\mathcal{C}} \overrightarrow{w}$$

heap of "half-edges" (i, j) for $\mathcal{C}$

"arrow"

$$\overrightarrow{w} = \begin{pmatrix} 1 & 1 & 1 & 2 & 2 & 3 & 3 & 3 & 3 \\ 3 & 1 & 2 & 3 & 1 & 3 & 3 & 1 & 2 \end{pmatrix}$$

biword

path on X

$$\omega = (s_0, \dots, s_i, s_{i+1}, \dots, s_n)$$

$$s_i \in X \quad i = 0, \dots, n$$

weight

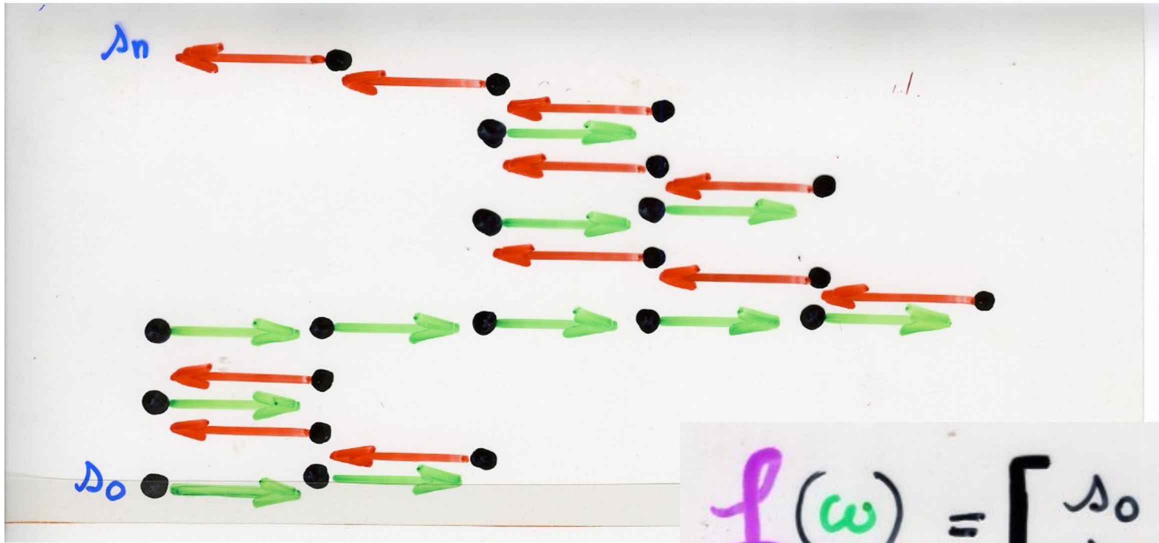
$$v(\omega) = \prod_{0 \le i \le n-1} v(s_i, s_{i+1})$$

$$X = [1, k]$$

$$a_{i,j} = v(i, j)$$

$$A = (a_{i,j})_{1 \le i,j \le k}$$

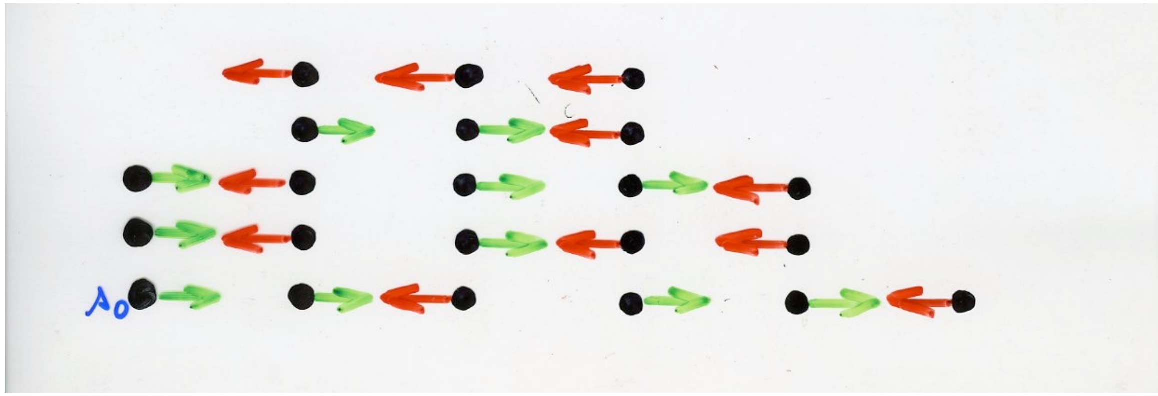
Path ω on X



$$f(\omega) = \begin{bmatrix} s_0 & s_1 & \cdots & s_{n-1} \\ s_1 & s_2 & \cdots & s_n \end{bmatrix}$$

commutation class

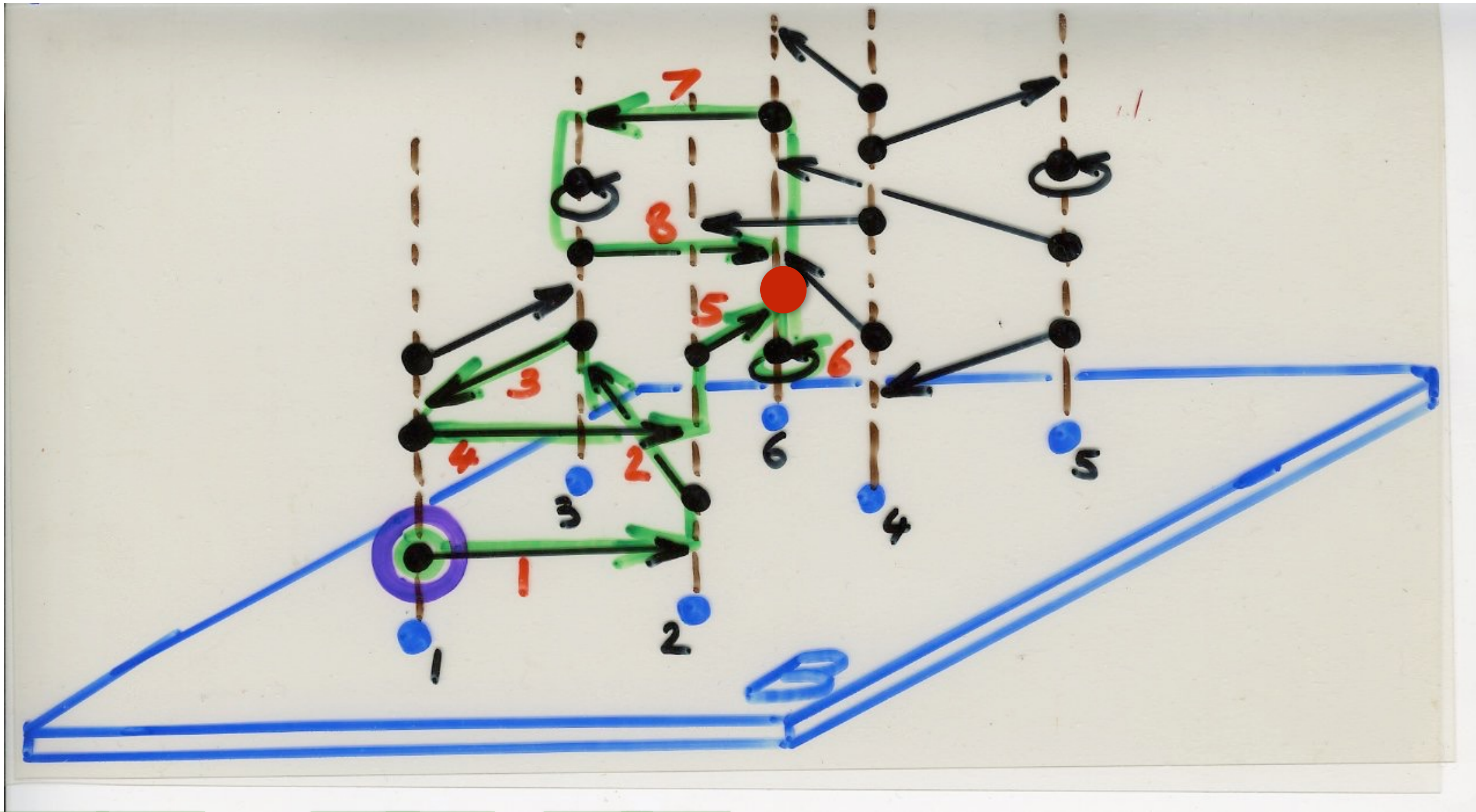
$\omega \longrightarrow f(\omega) \in F(X)$



$(s, \Phi) \xrightarrow{h} \omega$ path on X

algorithm "following" a flow $\Phi \in F(X)$

algorithm "following"
a flow $\Phi \in F(X)$



$$\begin{pmatrix} 1 & 1 & 1 \\ 2 & 2 & 3 \end{pmatrix} \begin{pmatrix} 2 & 2 \\ 3 & 6 \end{pmatrix} \begin{pmatrix} 3 & 3 & 3 \\ 1 & 6 & 3 \end{pmatrix} \begin{pmatrix} 4 & 4 & 4 & 4 \\ 6 & 2 & 5 & 6 \end{pmatrix} \begin{pmatrix} 5 & 5 & 5 \\ 4 & 6 & 5 \end{pmatrix} \begin{pmatrix} 6 & 6 \\ 6 & 3 \end{pmatrix}$$

$$h(1, \Phi) = (1, 2, 3, 1, 2, 6, 6, 3, 6)$$

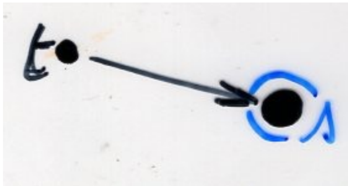
definition Φ flow $F(X)$

Φ rearrangement iff for any $s \in X$

$$\deg^{+}_{\Phi}(s) = \deg^{-}_{\Phi}(s)$$

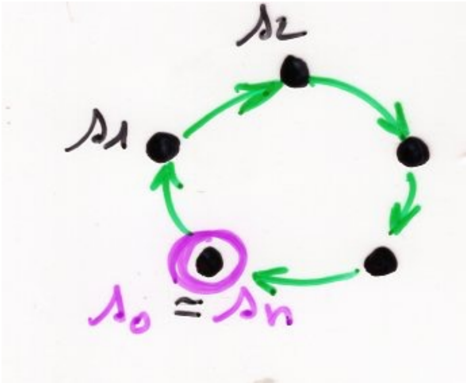
$$\deg^{+}_{\Phi}(s) = \left\{ \begin{array}{l} \text{number of edges } \binom{s}{t} \\ t \in X, \text{ in } \Phi \end{array} \right\}$$

$$\deg^{-}_{\Phi}(s) = \left\{ \begin{array}{l} \text{number of edges } \binom{t}{s} \\ t \in X \text{ in } \Phi \end{array} \right\}$$



$$\mathbf{R}(X) \subseteq \mathbf{F}(X)$$

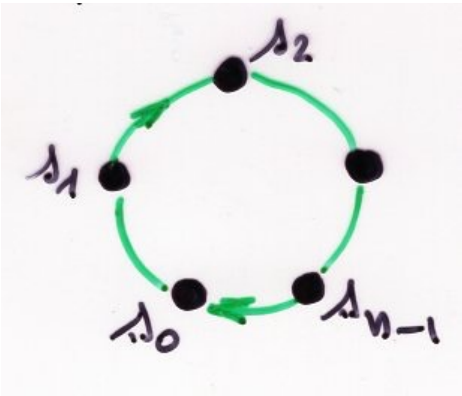
$R(X)$ submonoid of $F(X)$



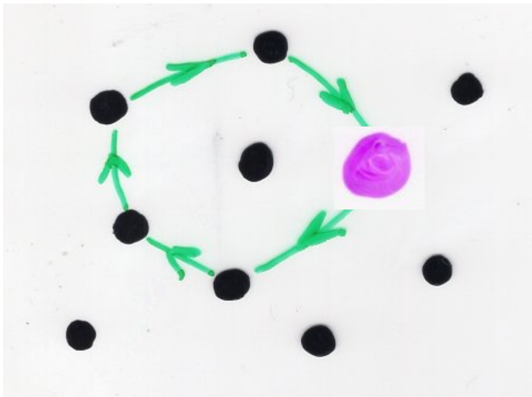
(from Chapter 2d)

here circuit = path $(s_0, \dots, s_n)$ with $s_0 = s_n$

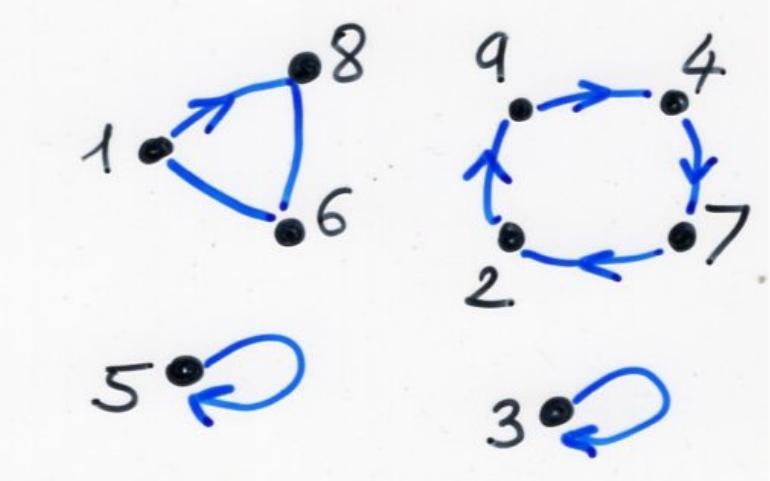
elementary circuit $\omega = (s_0, \dots, s_n)$ with $s_0 = s_n$, all vertices are disjoint except $s_0 = s_n$.



Cycle = elementary circuit up to a circular permutation of the vertices



pointed cycle



cycles of a permutation

sometimes

cycle = circuit
up to circular
permutation

our cycle
are called
elementary
cycle

heaps of cycles on X
monoid

$\mathbf{HC}(X)$

basic pieces : cycles on X

X

dependancy relation

$\gamma \, \mathcal{C} \, \gamma'$

iff $\mathrm{supp}(\gamma) \cap \mathrm{supp}(\gamma') \neq \emptyset$

$$E = \gamma_1 \odot \cdots \odot \gamma_k$$

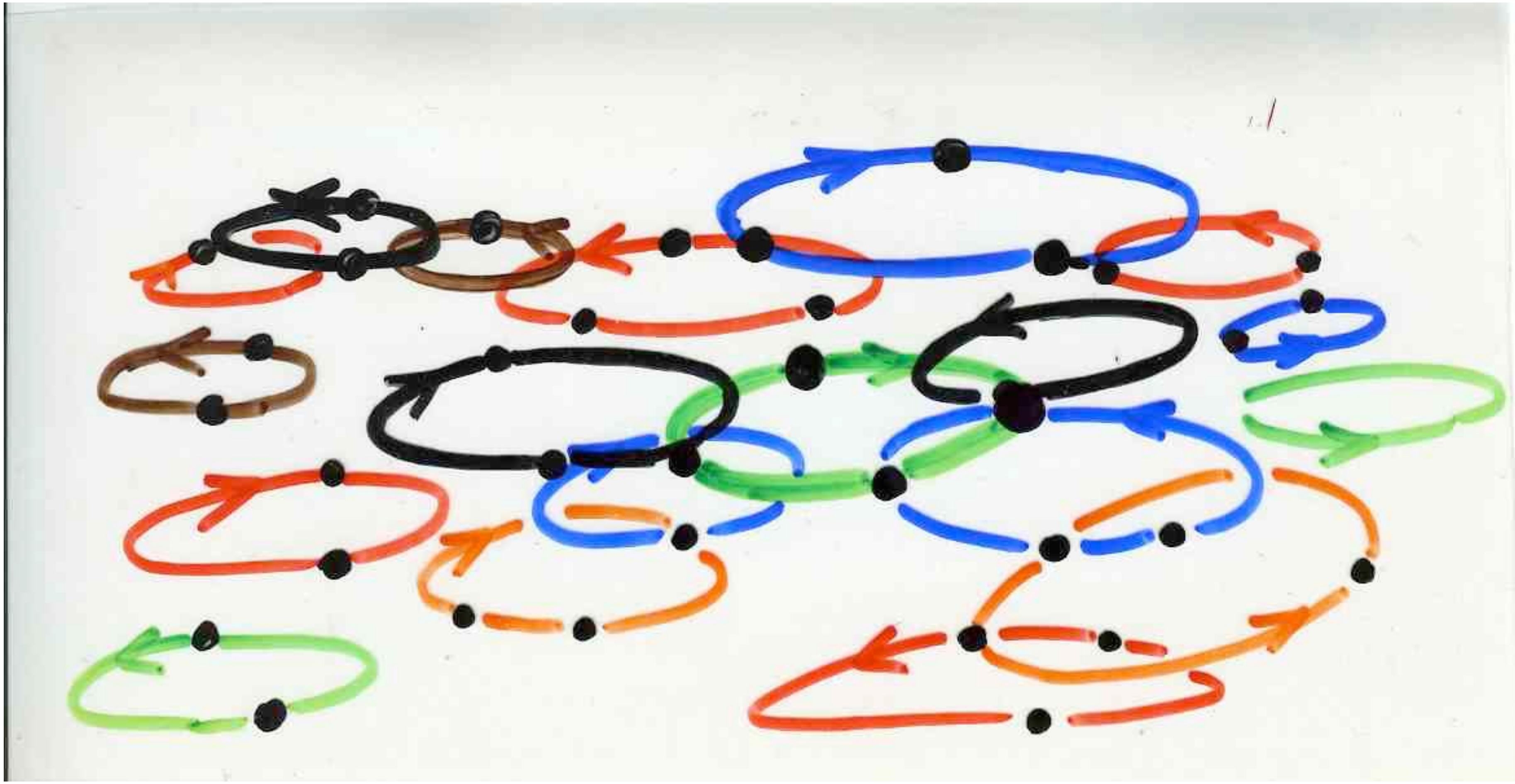
$$f(E) = f(\gamma_1) \odot \cdots \odot f(\gamma_k)$$

<u>Proposition</u> The map $f : \mathbf{HC}(X) \longrightarrow \mathbf{R}(X)$ is an isomorphism from the heaps of cycles monoid to the rearrangements monoid

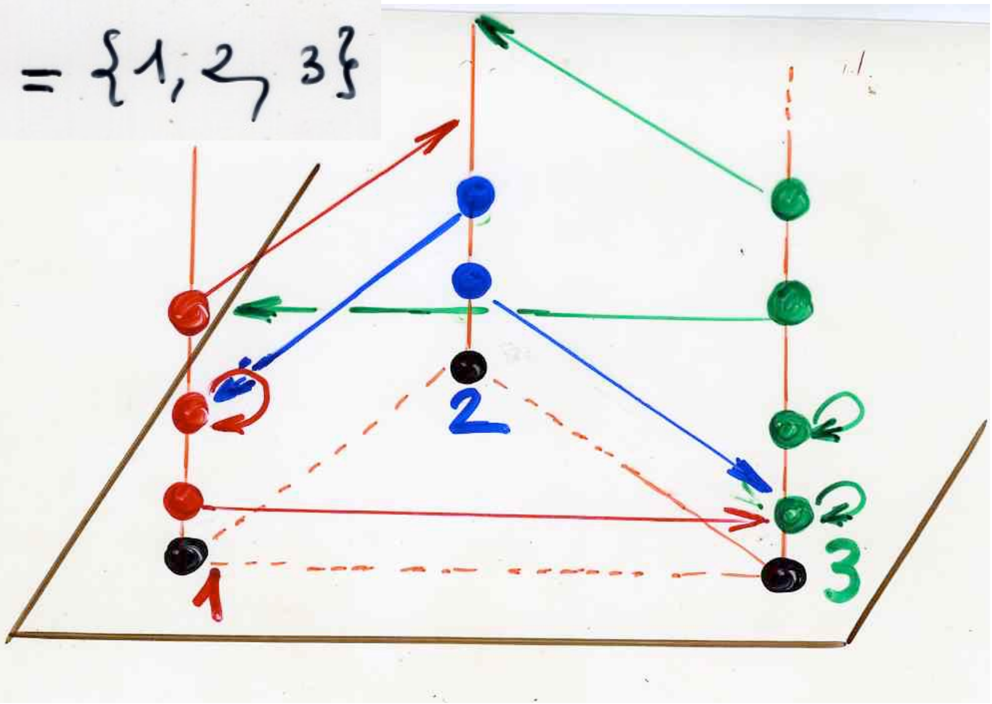
Construction of the reciprocal isomorphism

$$g = f^{-1}$$

$$HC(X) \simeq R(X)$$

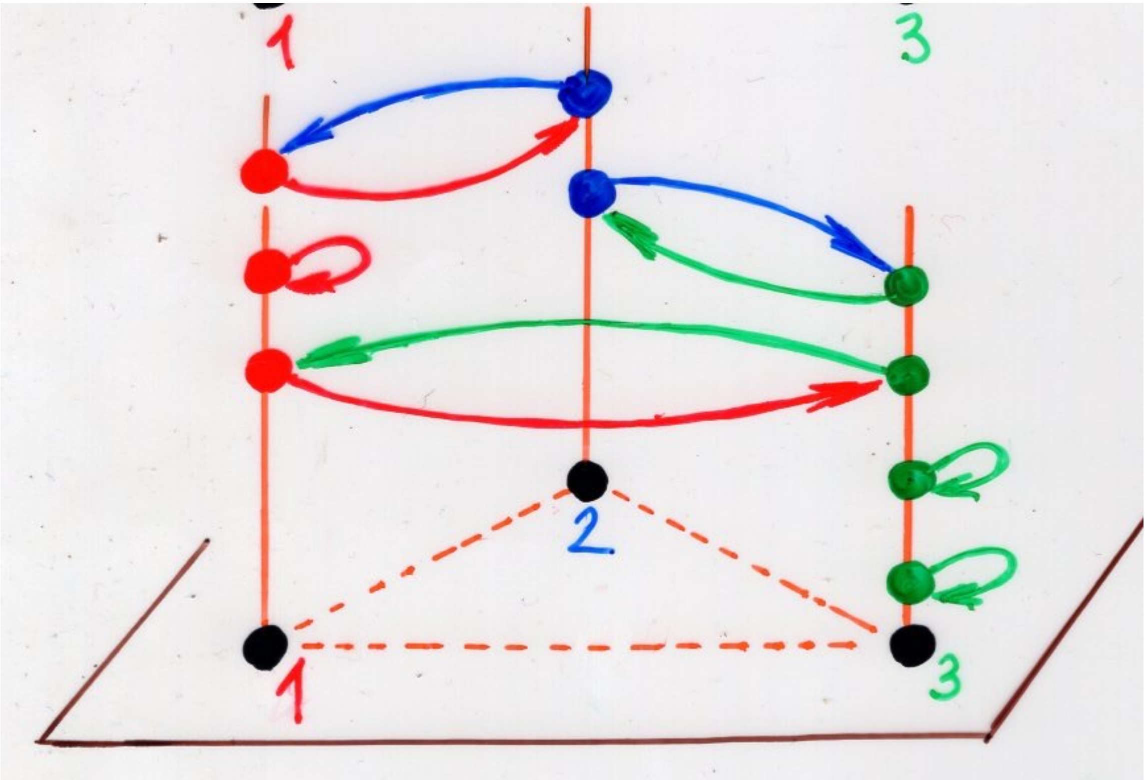


$X = \{1, 2, 3\}$



$$HC(X) \simeq R(X)$$

$$\begin{pmatrix} 1 & 1 & 1 & 2 & 2 & 3 & 3 & 3 & 3 \\ 3 & 1 & 2 & 3 & 1 & 3 & 3 & 1 & 2 \end{pmatrix}$$



Construction of the reciprocal isomorphism

$$g = f^{-1}$$

algorithm "following" a flow $\Phi \in F(X)$

variation of the proof

rearrangements = heaps of cycles

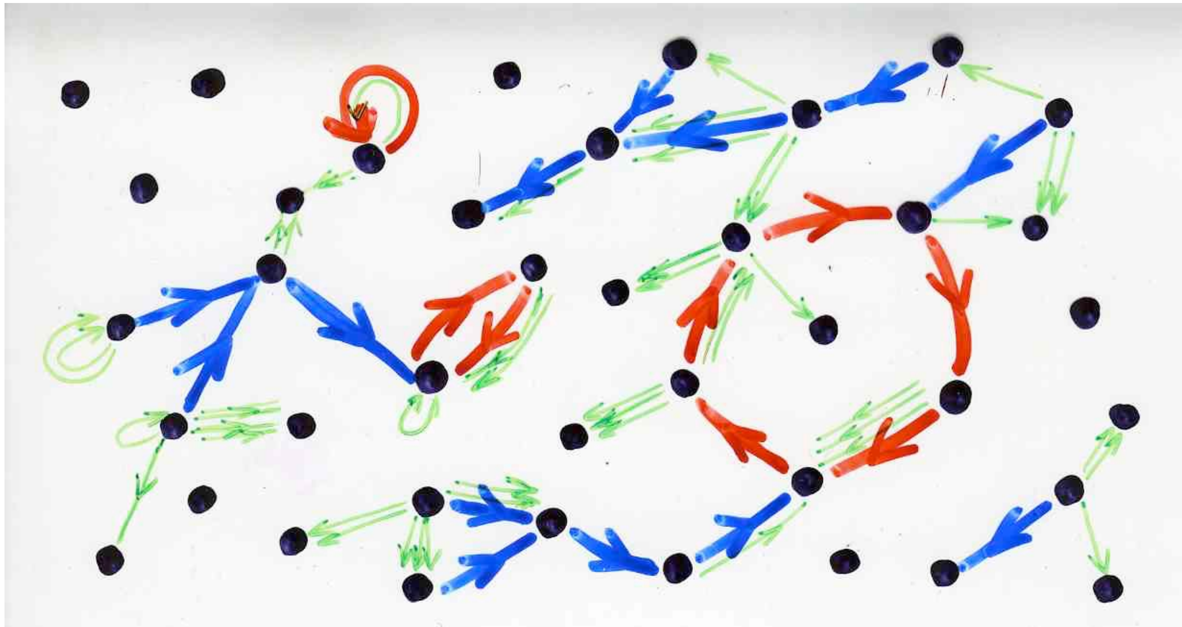
What do you "see" above:

- a (general) flow $F \in \mathcal{F}(X)$
- a rearrangement $\Phi \in \mathcal{R}(X)$

in other words:

describe the combinatorial structure made with the max (resp. min) oriented edges of the respective flow

above (or below)
a flow $F \in \mathcal{F}(X)$

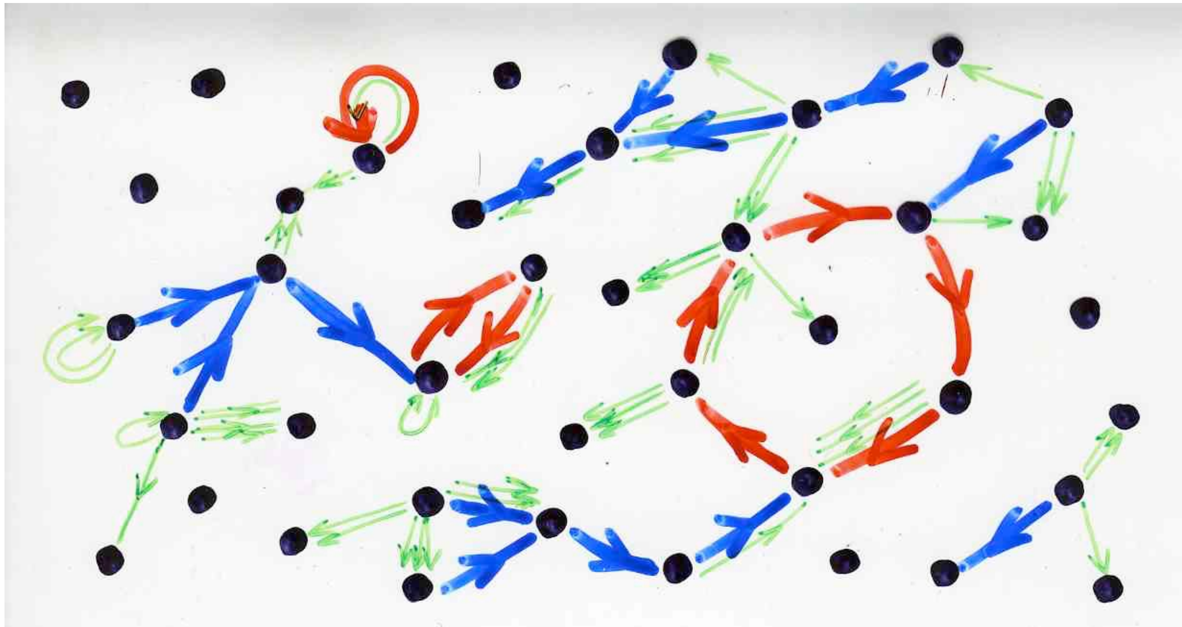


$Y = \mathrm{supp}(F)$
support

the max edges of F formed
an endofonction of Y
i.e. a map $\varphi : Y \to Y$

$Y \subseteq X$ set of vertices $s \in X$
covered by the flow F
i.e. $\exists t$, such that $(s, t) \in F$

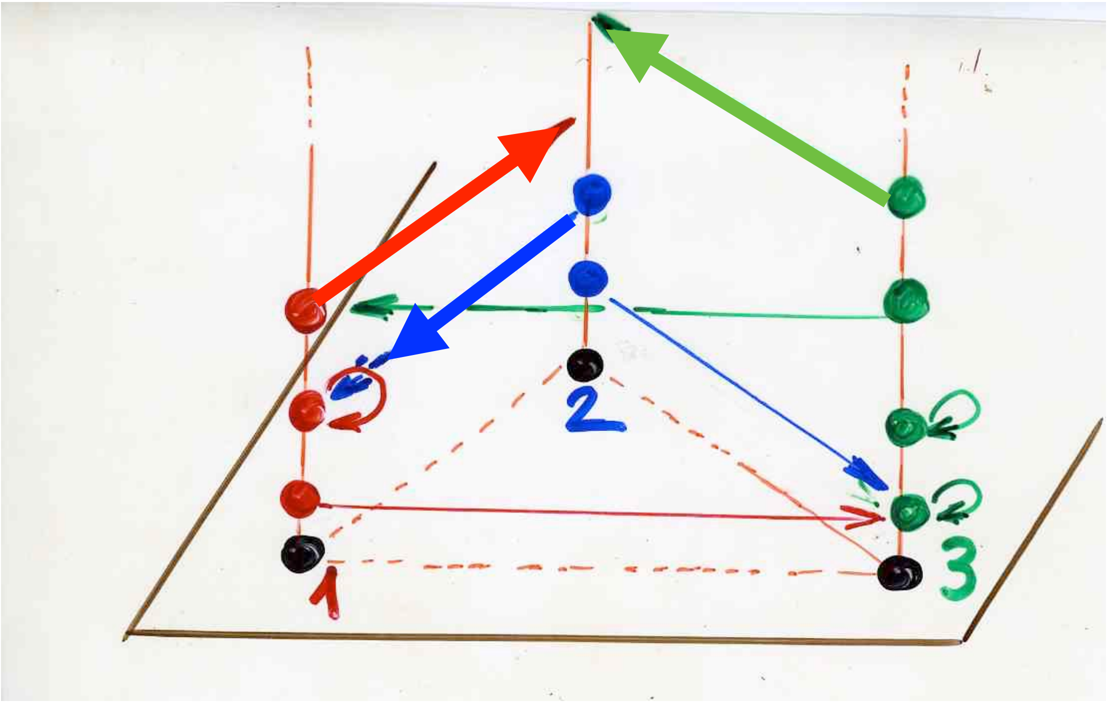
above (or below)
a flow $F \in F(X)$

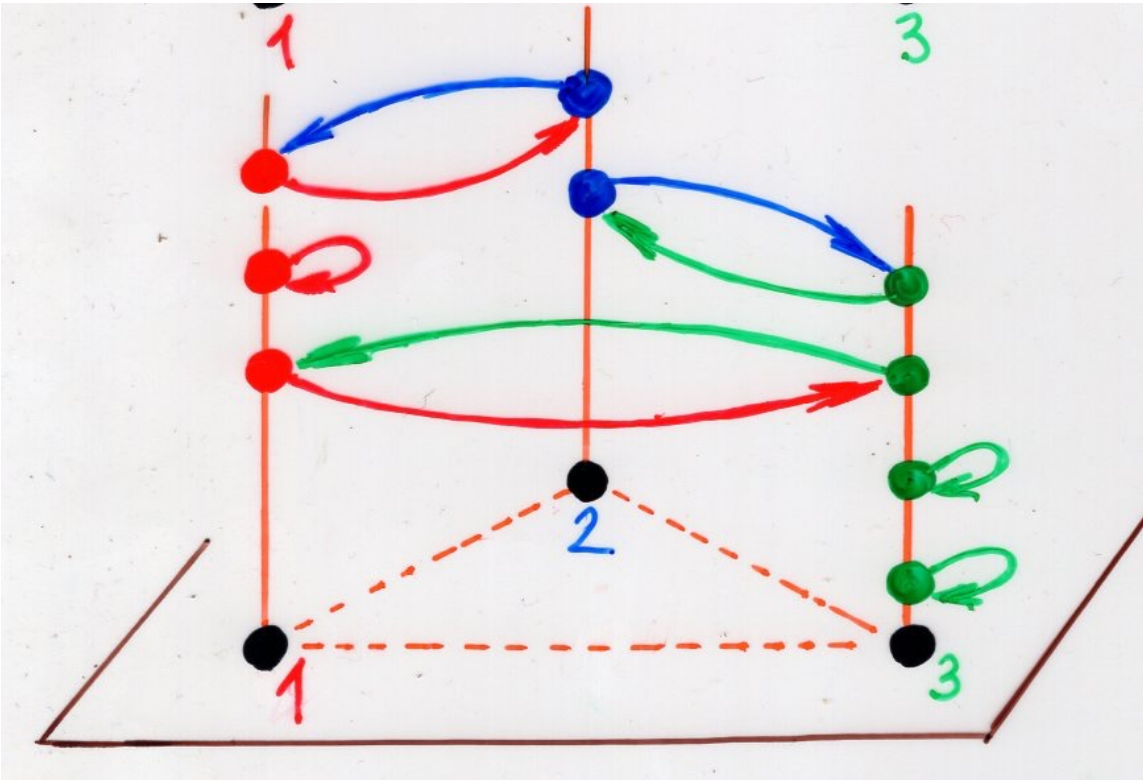


the max edges of F formed
an endofonction of Y
i.e. a map $\varphi : Y \to Y$

correction to the slide in the video of the course:

if $F \in R(X)$
there exist at least
one cycle in the
endofonction φ





$X = \{1, 2, 3\}$

$$\begin{pmatrix} 1 & 1 & 1 & 2 & 2 & 3 & 3 & 3 & 3 \\ 3 & 1 & 2 & 3 & 1 & 3 & 3 & 1 & 2 \end{pmatrix}$$

Construction of the reciprocal isomorphism

$g = f^{-1}$

algorithm "following" a flow $\Phi \in F(X)$

$F \in R(X)$ $\quad F = \Phi \circ \gamma$

rearrangement $\quad$ cycle

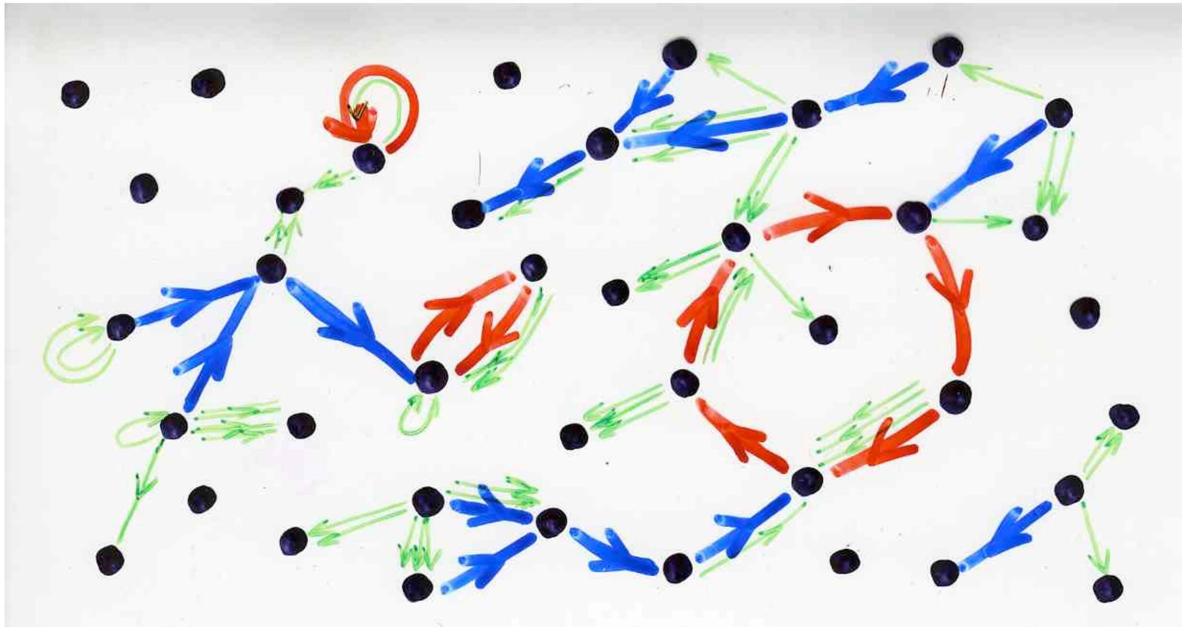
$g(F) = \gamma_1 \circ \cdots \circ \gamma_k \in HC(X)$ $\quad f(\gamma_1) \cdots f(\gamma_k) = F$

$$HC(X) \simeq R(X)$$

$g = f^{-1}$

heaps of cycles monoid on X $\quad$ rearrangements monoid on X

remark on the species endofunction



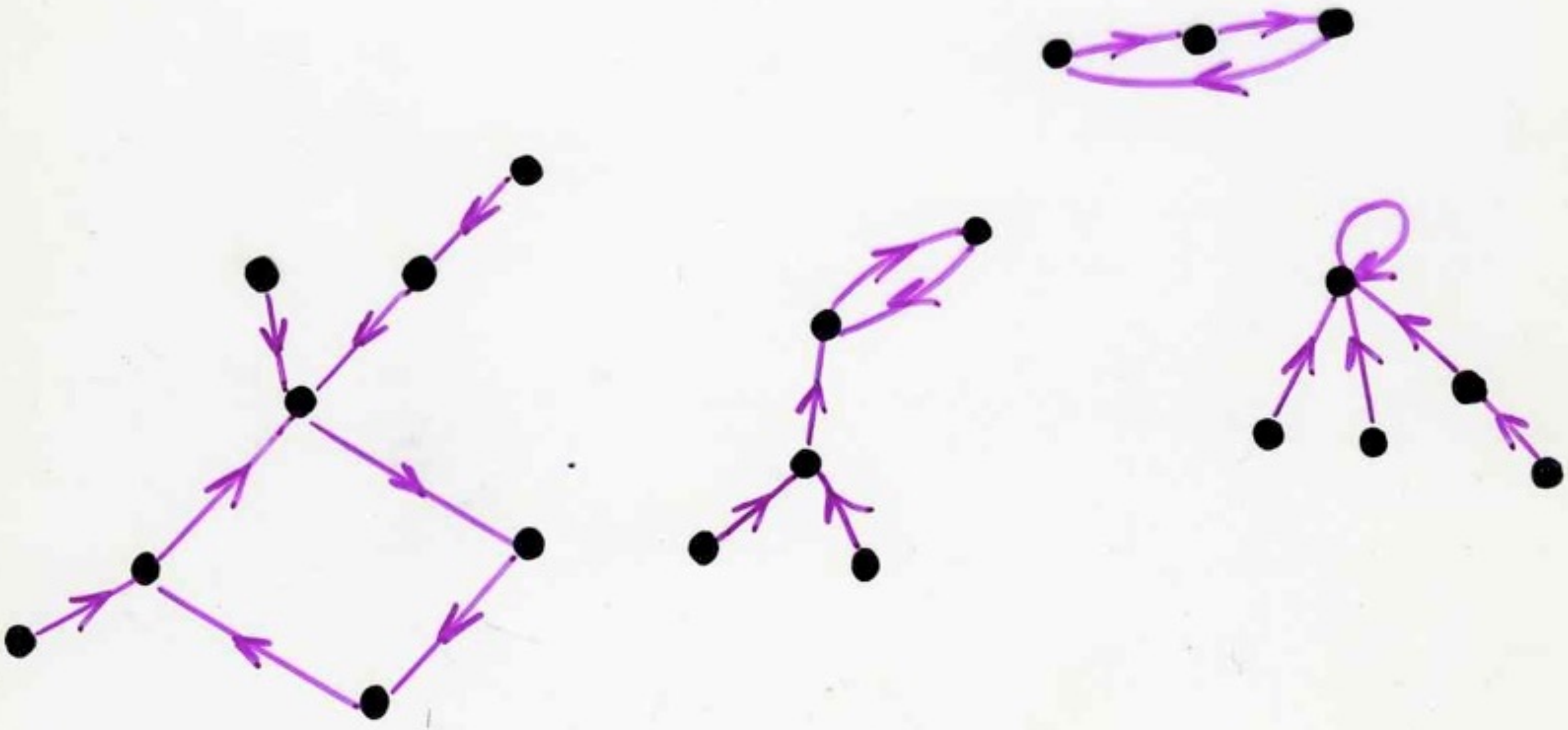
the max edges of F formed
an endofonction of Y
i.e. a map $\varphi : Y \to Y$

as species :
substitution
of the species
arborescence
into the species
permutation

arborescence
= pointed
trees
("Cayley trees")
n^{n-2}

ex. Endofonctions

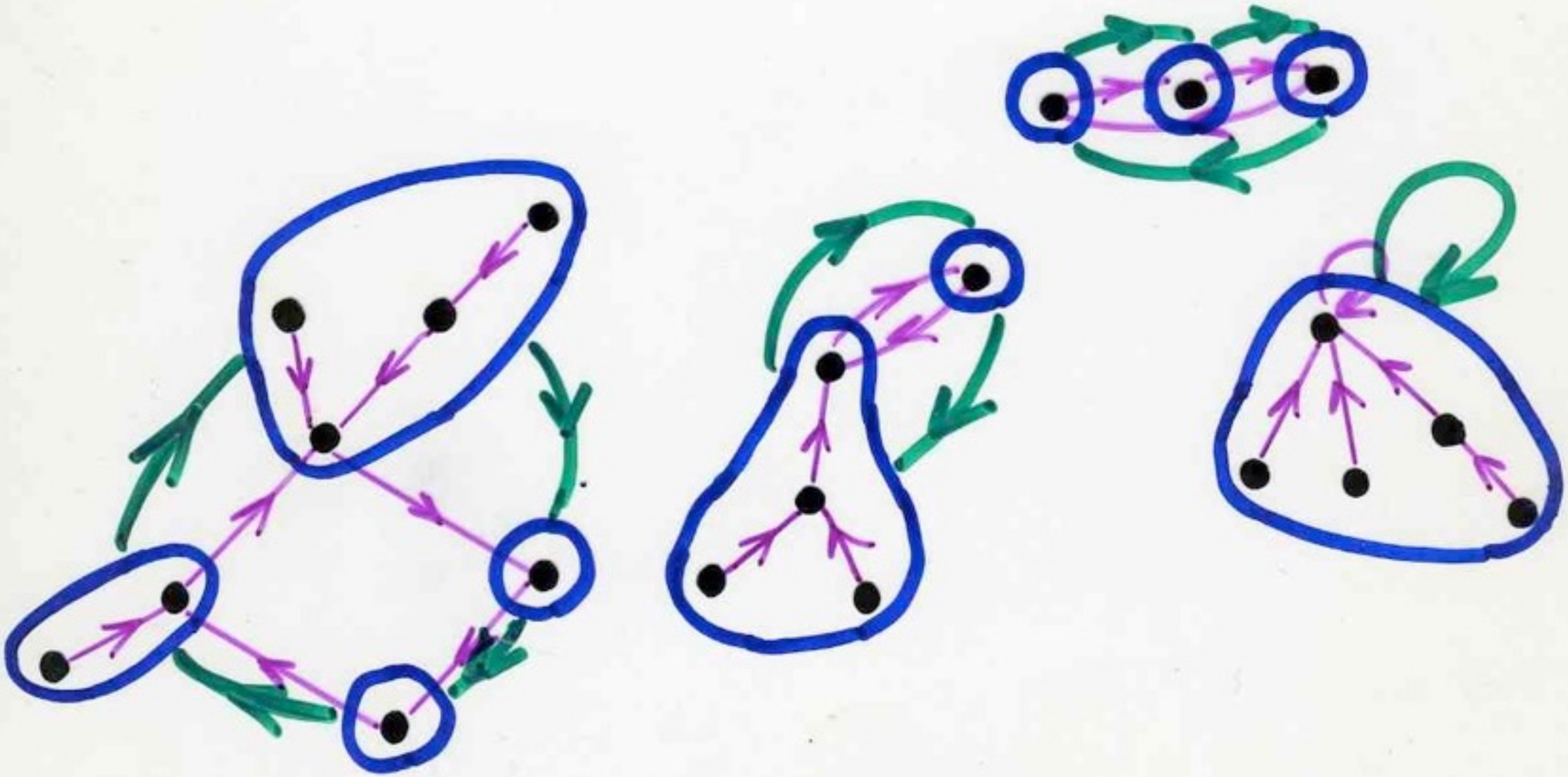
$$\underbrace{End}_{\text{endofonctions}} = \underbrace{S}_{\text{permutations}} \circ \underbrace{A}_{\text{arborescences}}$$



ex. Endofonctions

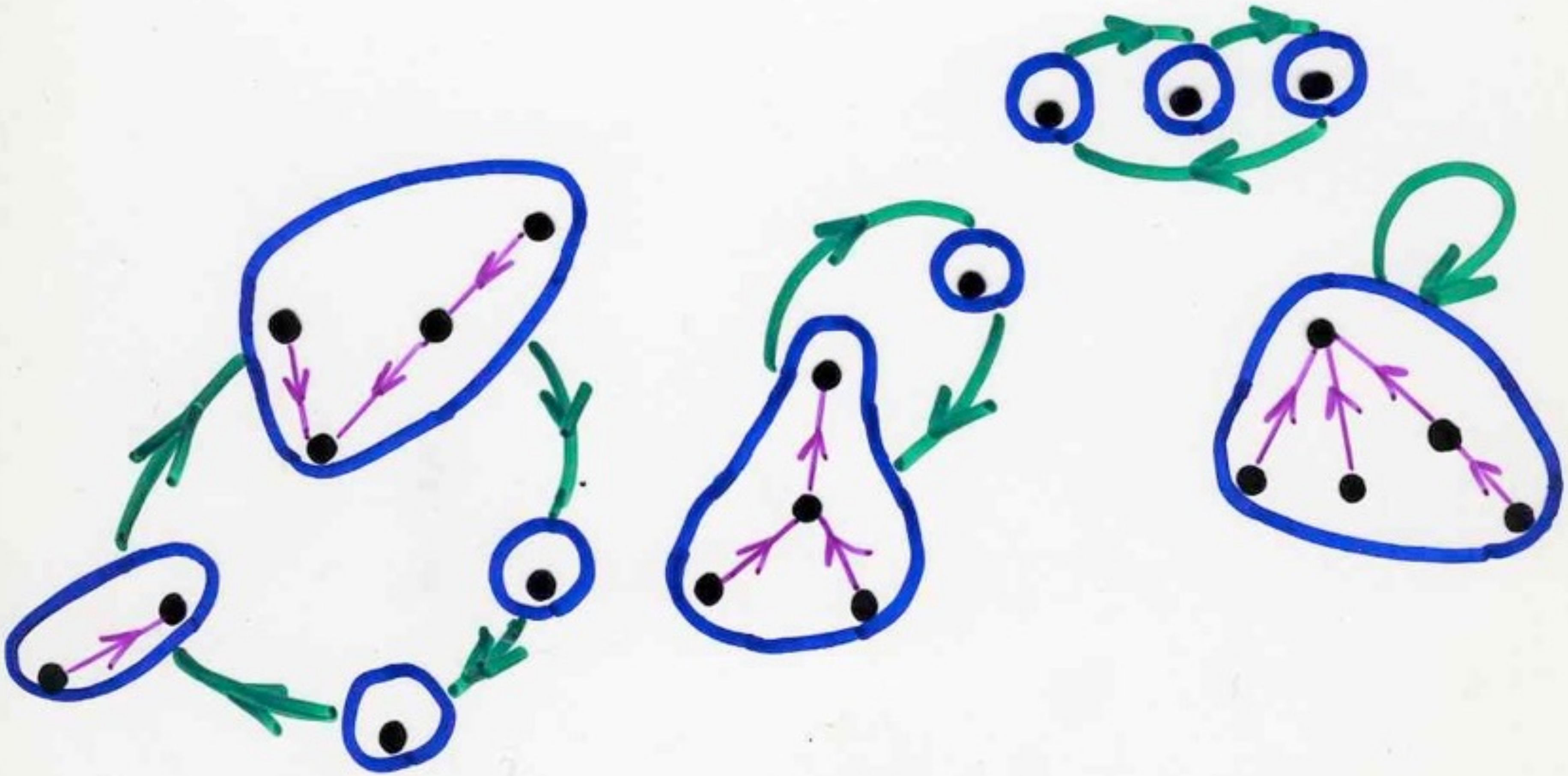
$$\mathrm{End} = \underset{\text{permutations}}{S} \circ \underset{\text{arborescences}}{A}$$

endofonctions



ex. Endofonctions

$$\underset{\text{endofonctions}}{End} = \underset{\text{permutations}}{S} \circ \underset{\text{arborescences}}{A}$$



exercise

Prove that every commutation monoid is isomophic to a submonoid of ~~the~~ a rearrangement monoid $R(X)$

paths and heaps of cycles

Bijection

$u, v \in X$

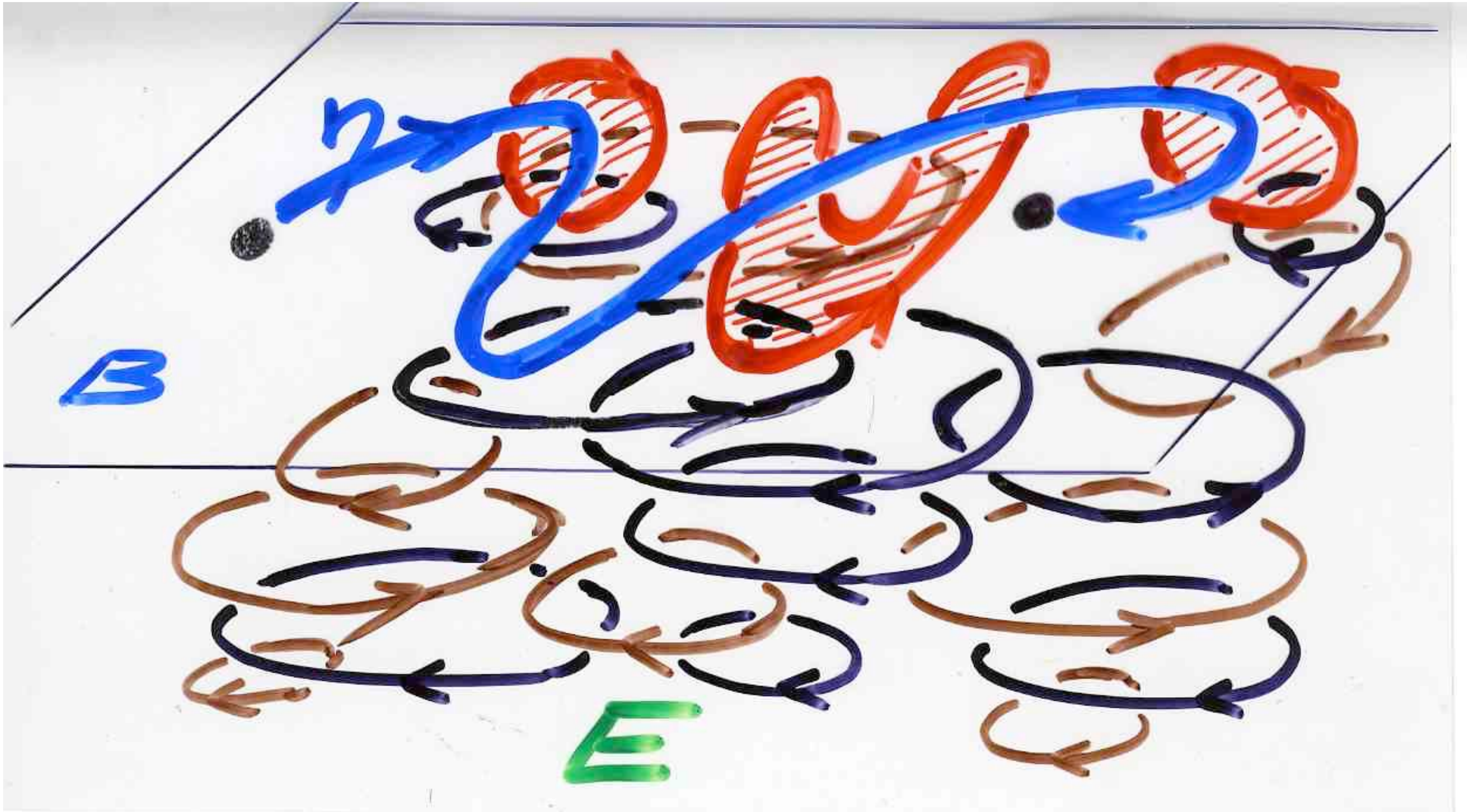
path ω on X going from u to v $\xleftrightarrow{\chi}$ (η, E)

- η self-avoiding path going from u to v
- E heap of cycles such that the projections $\alpha = \pi(m)$ of the maximal pieces intersect η ($\underset{\text{cycle}}{\alpha}$ and $\underset{\text{path}}{\eta}$ has a common vertex)

for any $s, t \in X$
the numbers of occurrences of the edge (s,t) in ω and in (η, E) are the same.

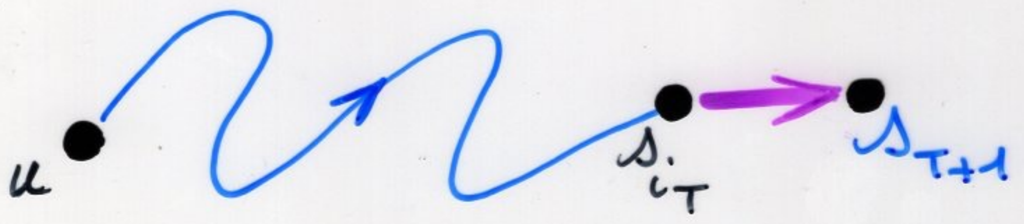
$$\Rightarrow \quad v(\omega) = v(\eta)\, v(E)$$

The bijection X



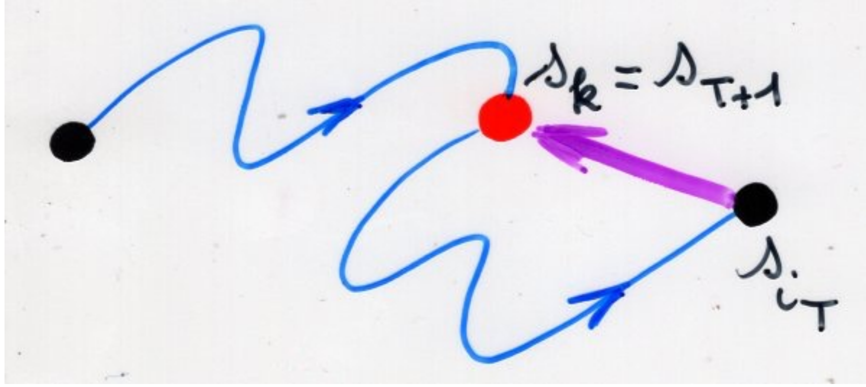
• suppose $\begin{cases} Cut_T(\omega) = (s_0 = u, \ldots, s_{i_T}) \\ E_T(\omega) \text{ heap of cycles} \end{cases}$

(i) if $s_{T+1} \notin Cut_T(\omega)$



$$\begin{cases} Cut_{T+1}(\omega) = (s_0 = u, \ldots, s_{i_T}, s_{T+1}) \\ E_{T+1}(\omega) = E_T(\omega) \end{cases}$$

(ii) if $s_{T+1} \in Cut_T(\omega)$, $s_{T+1} = s_k$



$$\begin{cases} Cut_{T+1}(\omega) = (s_0 = u, \dots, s_k) \\ E_{T+1}(\omega) = E_T(\omega) \odot \gamma \end{cases}$$

$$\dot{\gamma} = (s_k, \dots, s_{i_T}, s_{T+1} = s_k)$$

$$\omega \xrightarrow{\chi} (\eta, E)$$

$$\eta = \mathrm{Cut}_n(\omega)$$

$$E = E_n(\omega)$$

loop-erased process

LERW

Lawler, 1987

$$\omega \longrightarrow \left(\eta \, ; \, (\dot{\gamma}_1, \dots, \dot{\gamma}_{r_n}) \right)$$

self-avoiding path $u \rightsquigarrow v$

sequence of pointed cycles

from the pair $(\eta \, ; (\dot{\gamma}_1, \dots ; \dot{\gamma}_{r_n}))$ we can reconstruct the path ω

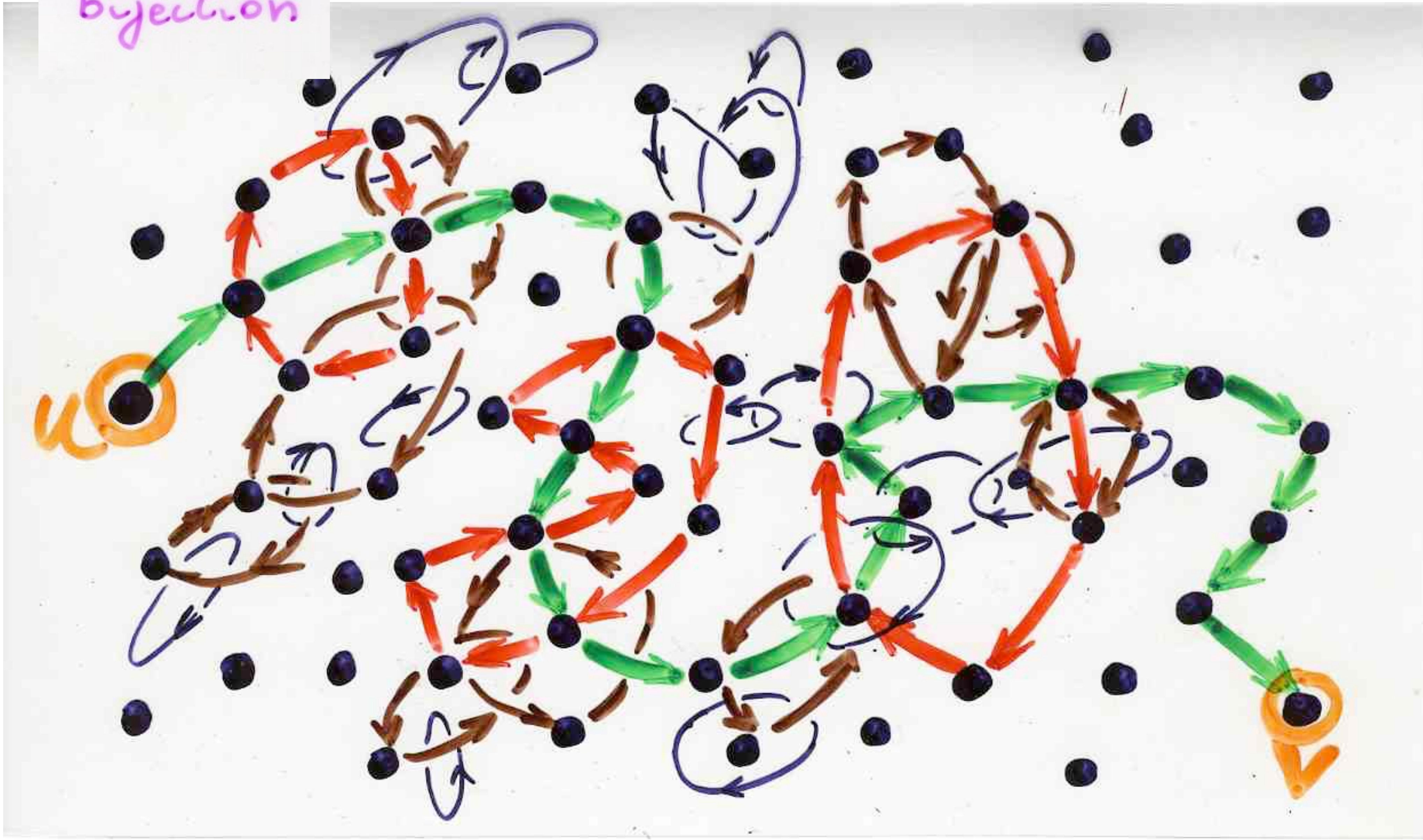
$$(\dot{\gamma}_1, \dots, \dot{\gamma}_{r_n}) \longrightarrow E = \gamma_1 \odot \cdots \odot \gamma_{r_n}$$

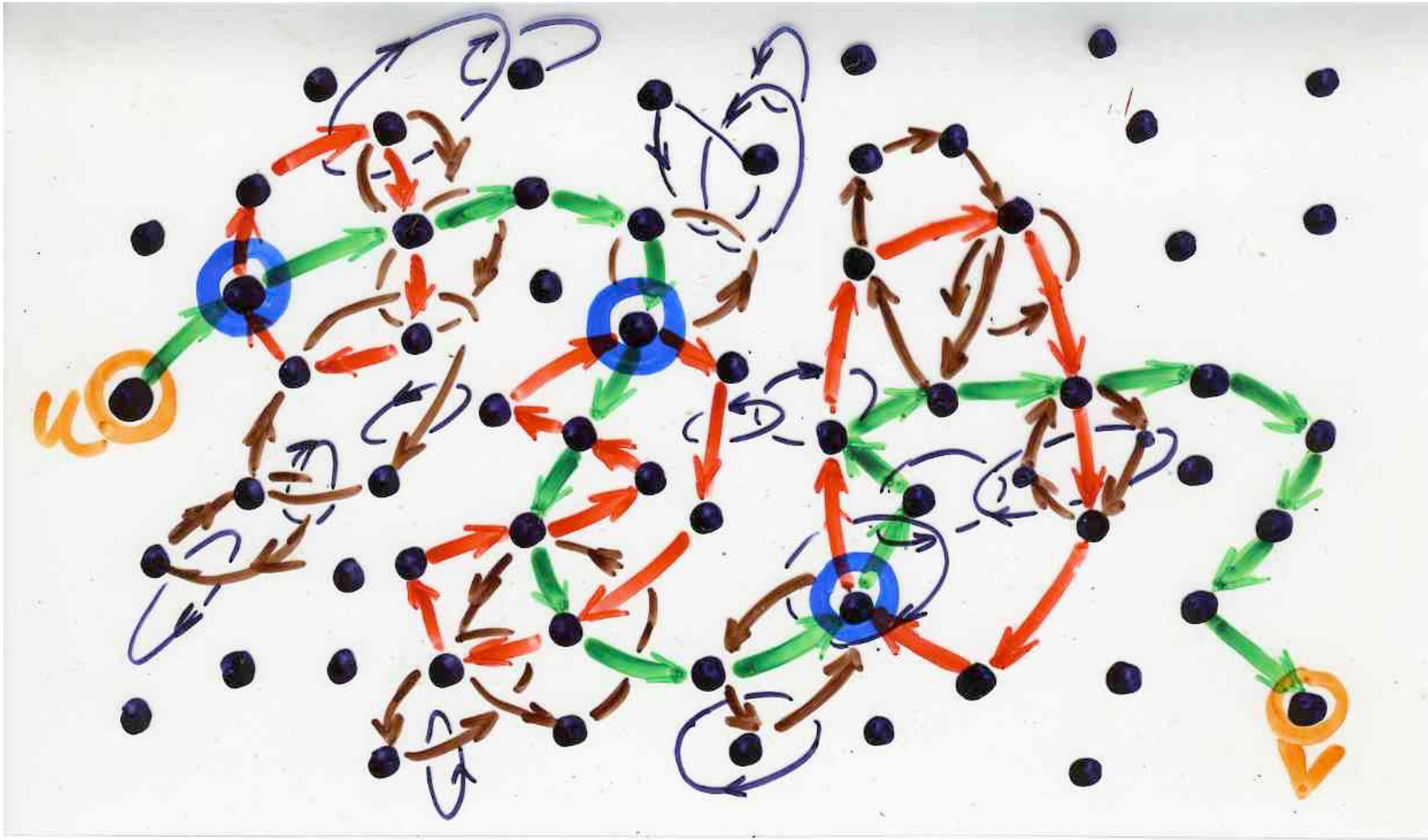
$$\omega \longrightarrow (\eta, E)$$

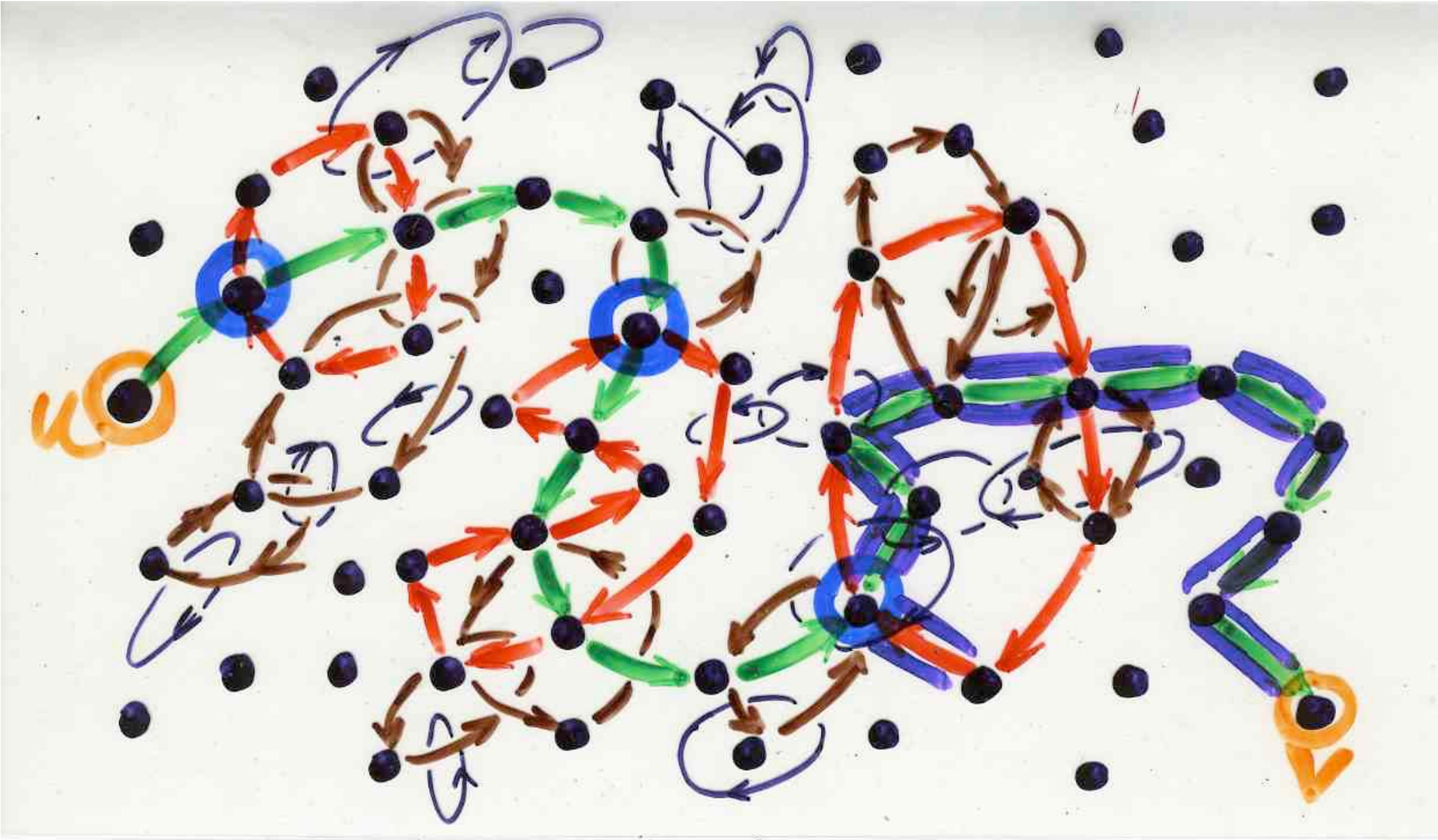
heaps of cycles on X

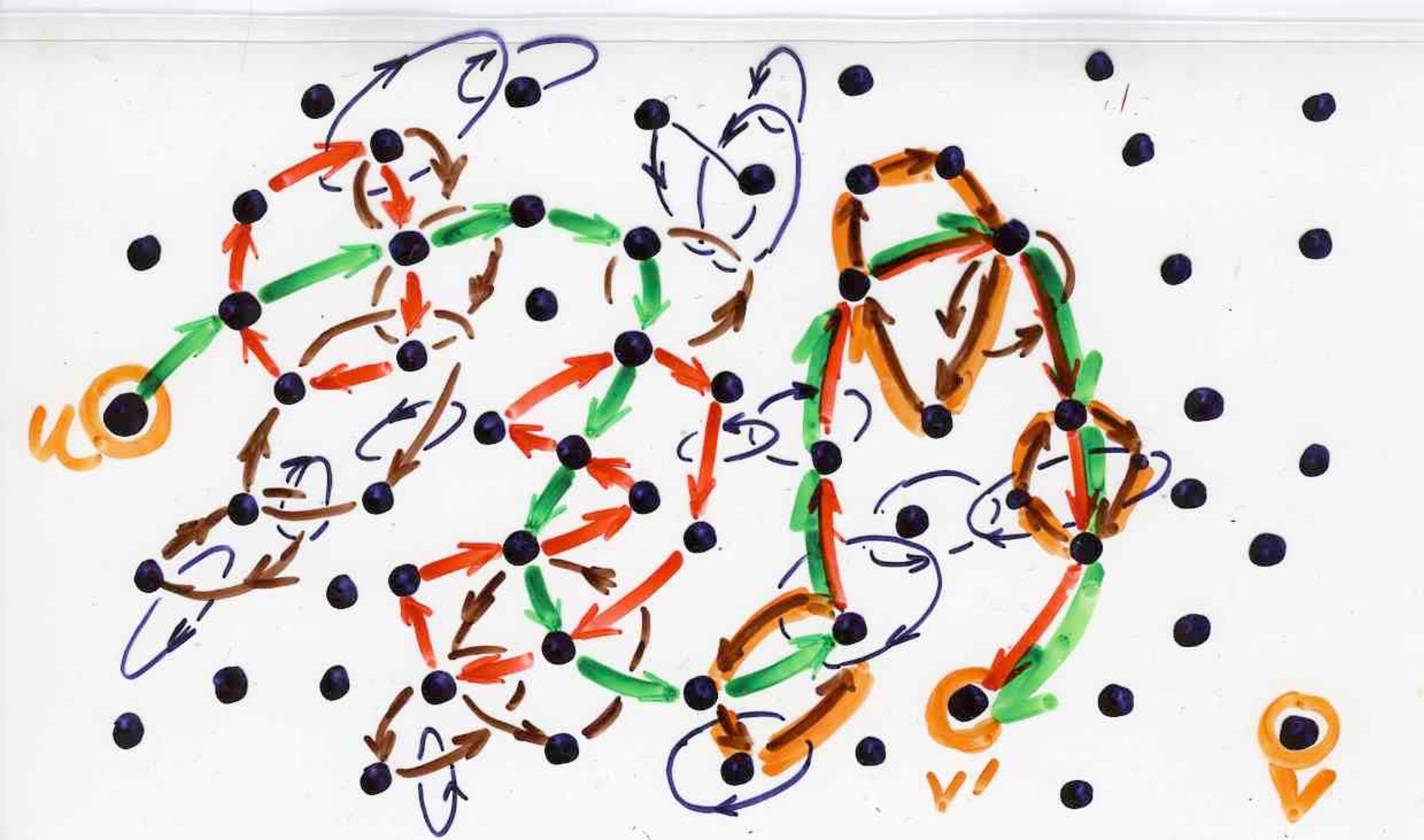
monoid

reverse
bijection




u
v


u
v



u
v'
v

$$
\begin{tikzcd}
\omega \arrow[r, "\chi"] \arrow[d] & (\eta, E) \arrow[d] \\
f(\omega) & f(E) \odot f(\eta)
\end{tikzcd}
$$

Lemma

$$f(\omega) = f(E)\, f(\eta)$$

"following" the flow $f(E) \odot f(\eta)$, starting at s_0, gives back ω

"breaking" paths and heap of cycles

second bijection

"gluing" bijections

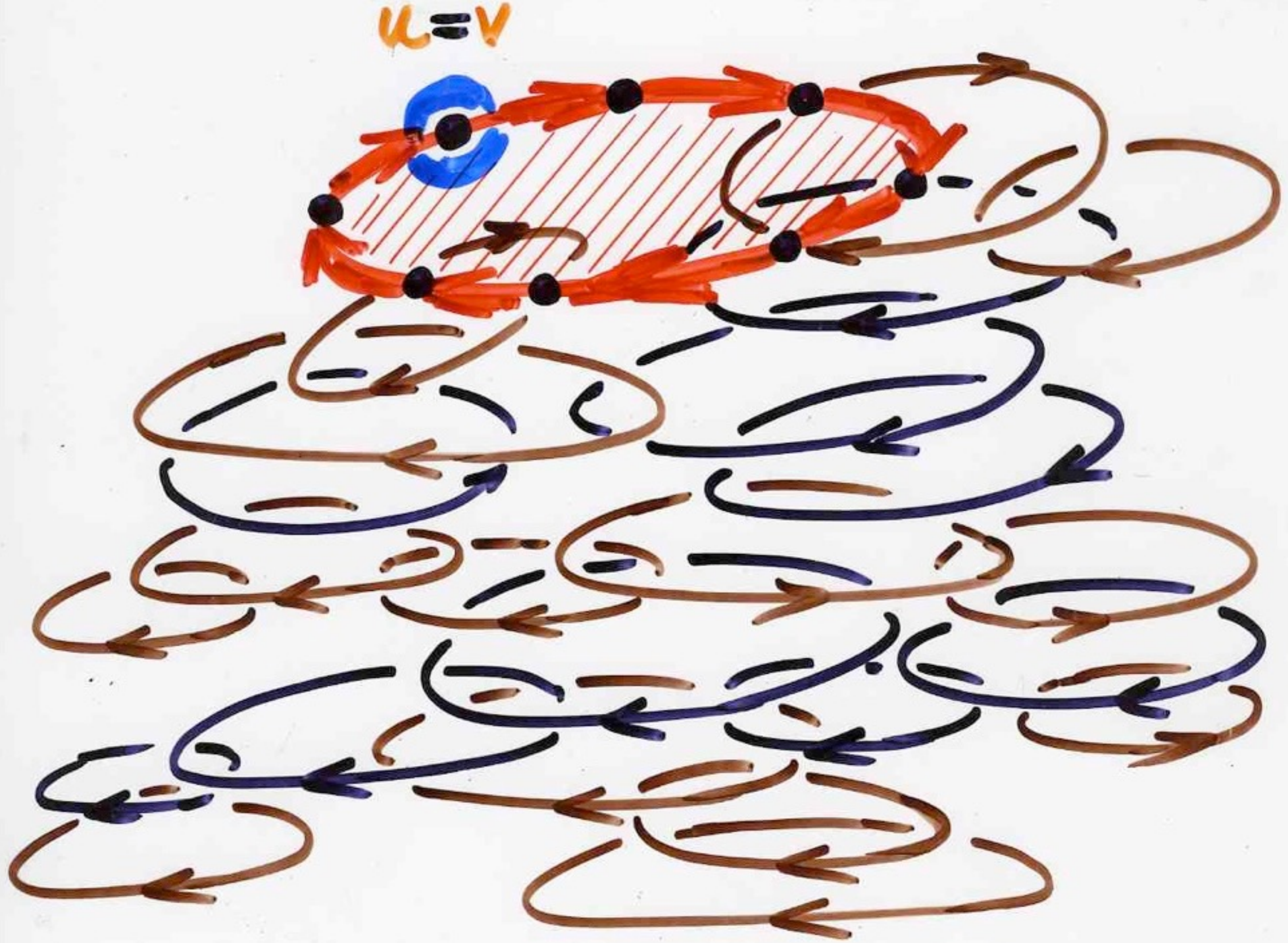
Circuit
path $\omega = (s_0, .., s_n)$ with $s_n = s_0$

Corollary Circuits on X are in bijection with pointed pyramids of cycles

= the unique cycle maximal piece has a distinguished vertex (or edge)

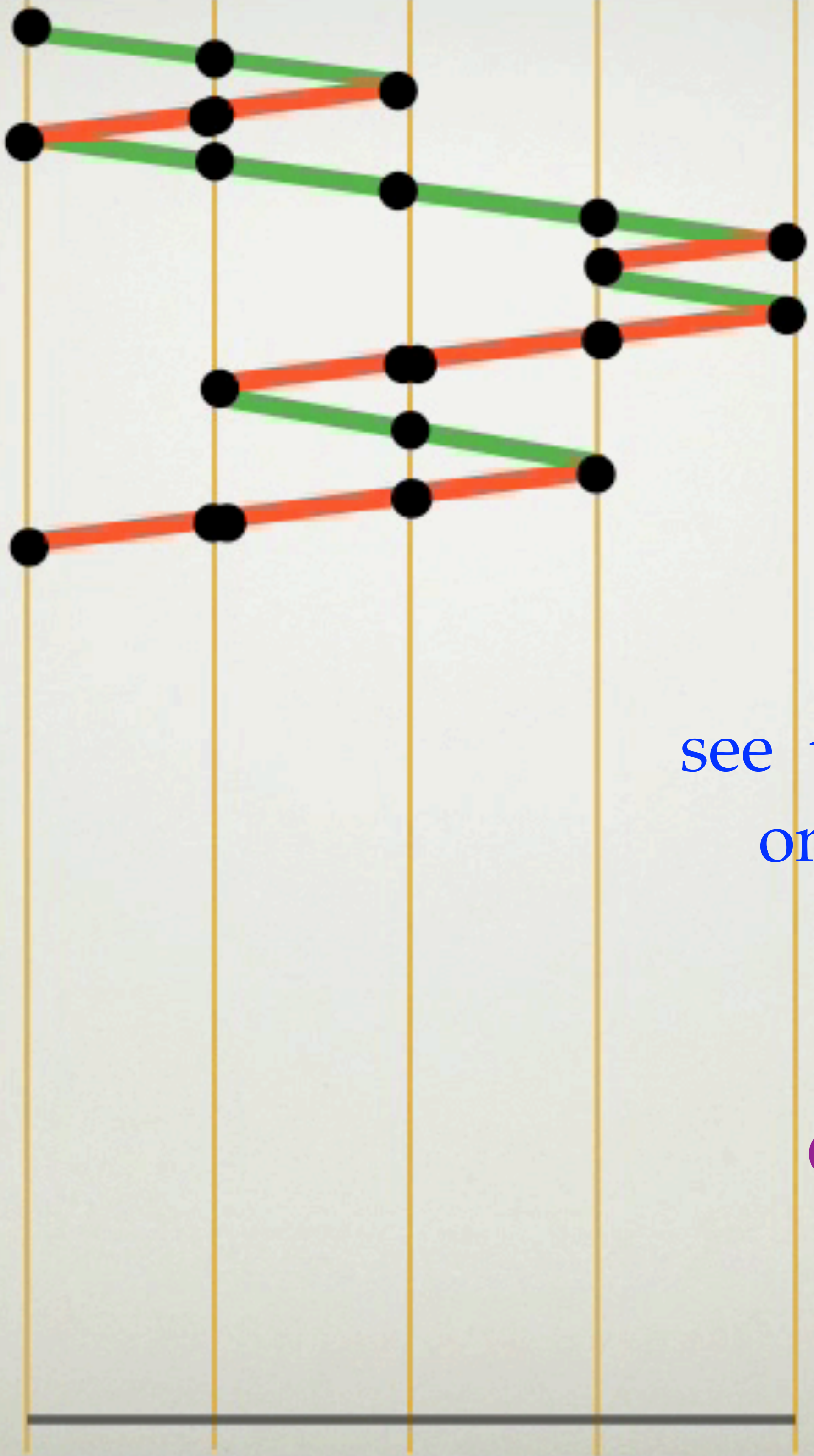
η is reduced to the vertex $u = v$

The bijection X for circuits $u = v$



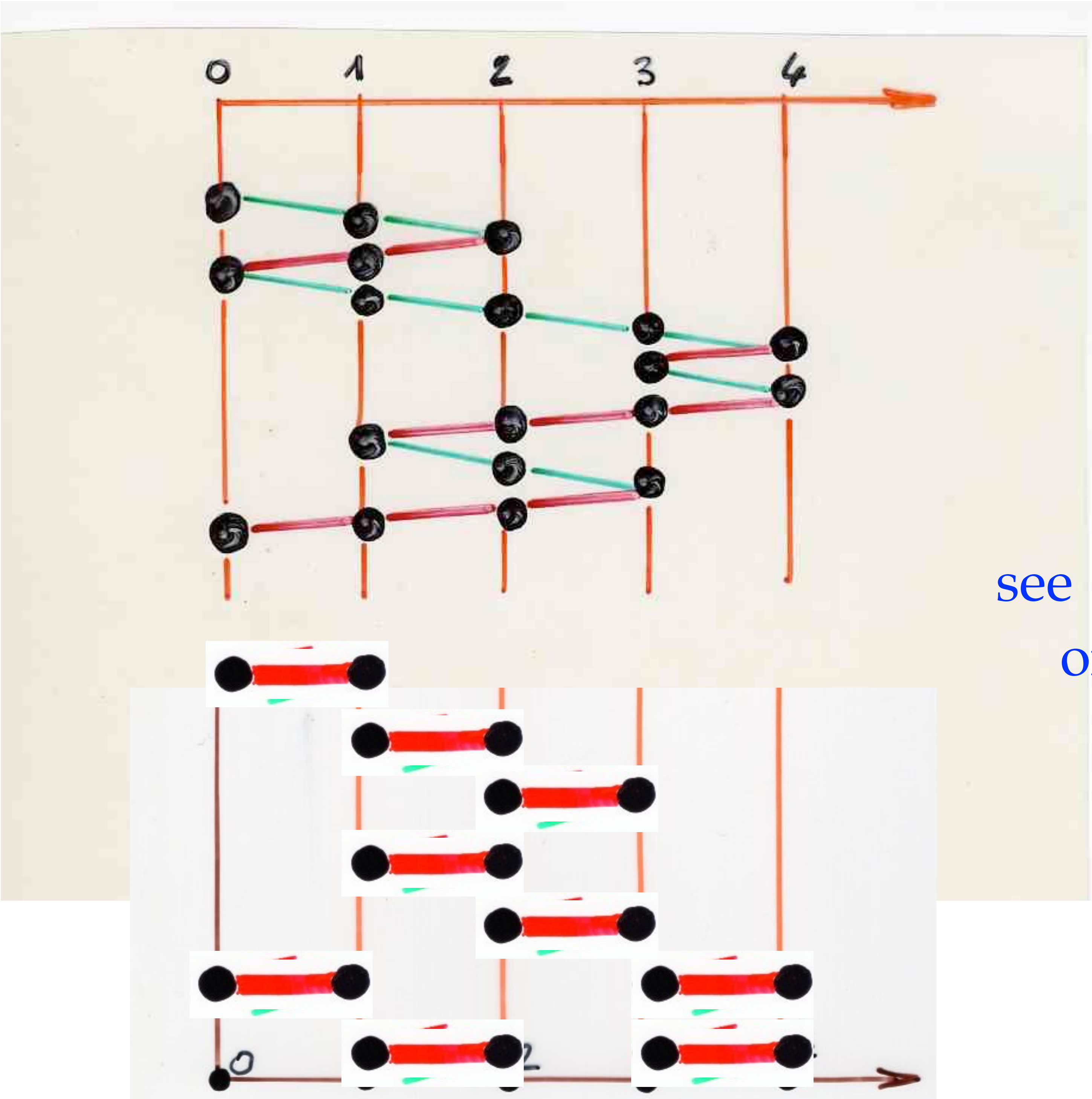
pointed cycles pyramid

an example with Dyck paths



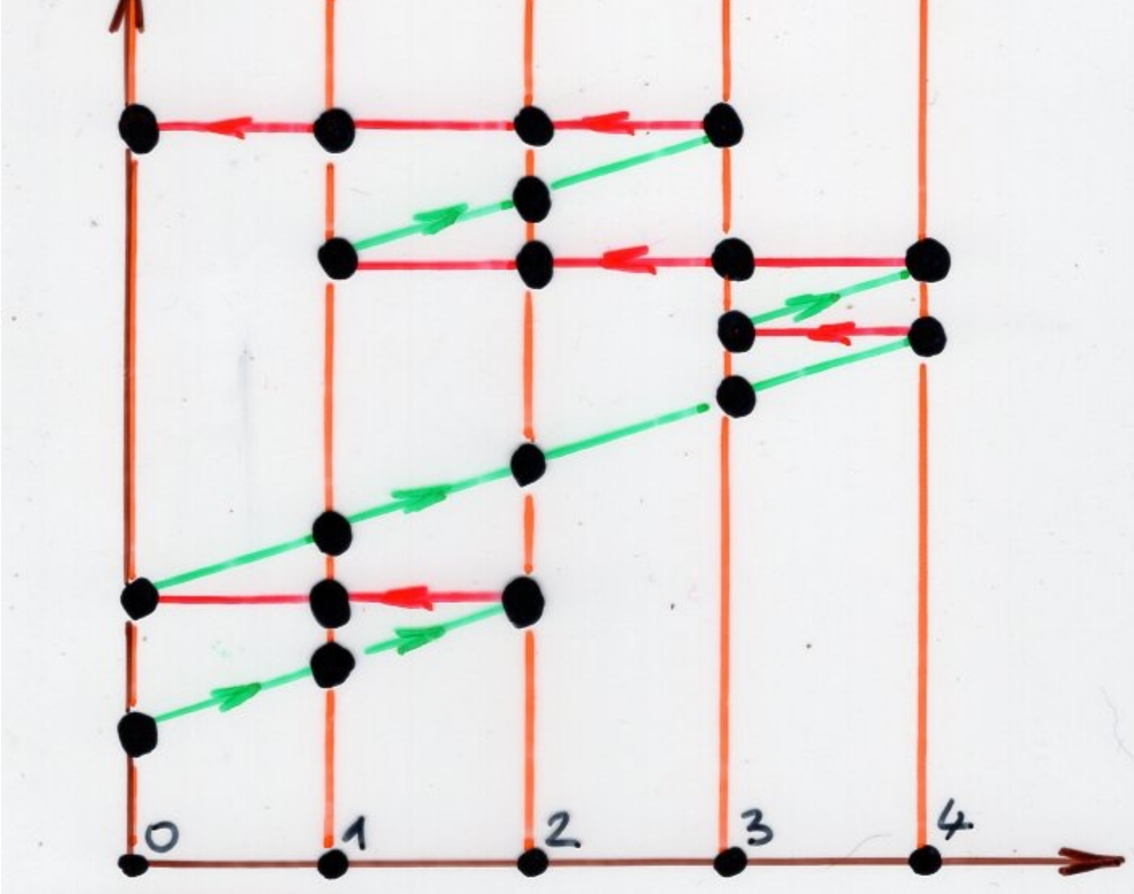
see the animation
on the video

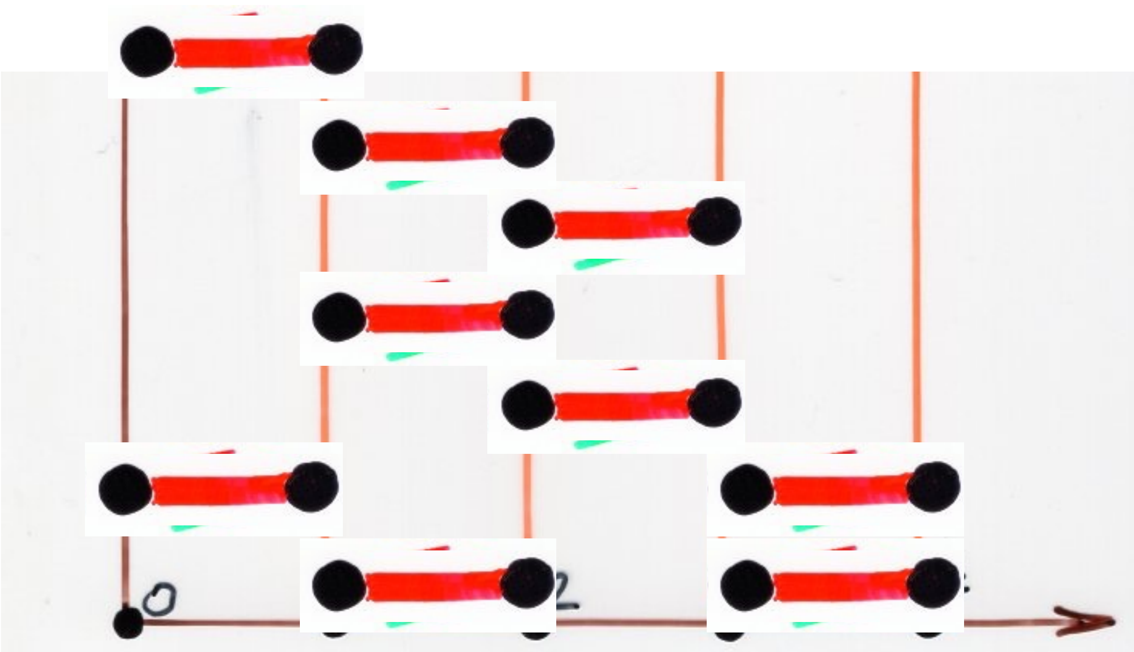
violin:
G. Duchamp

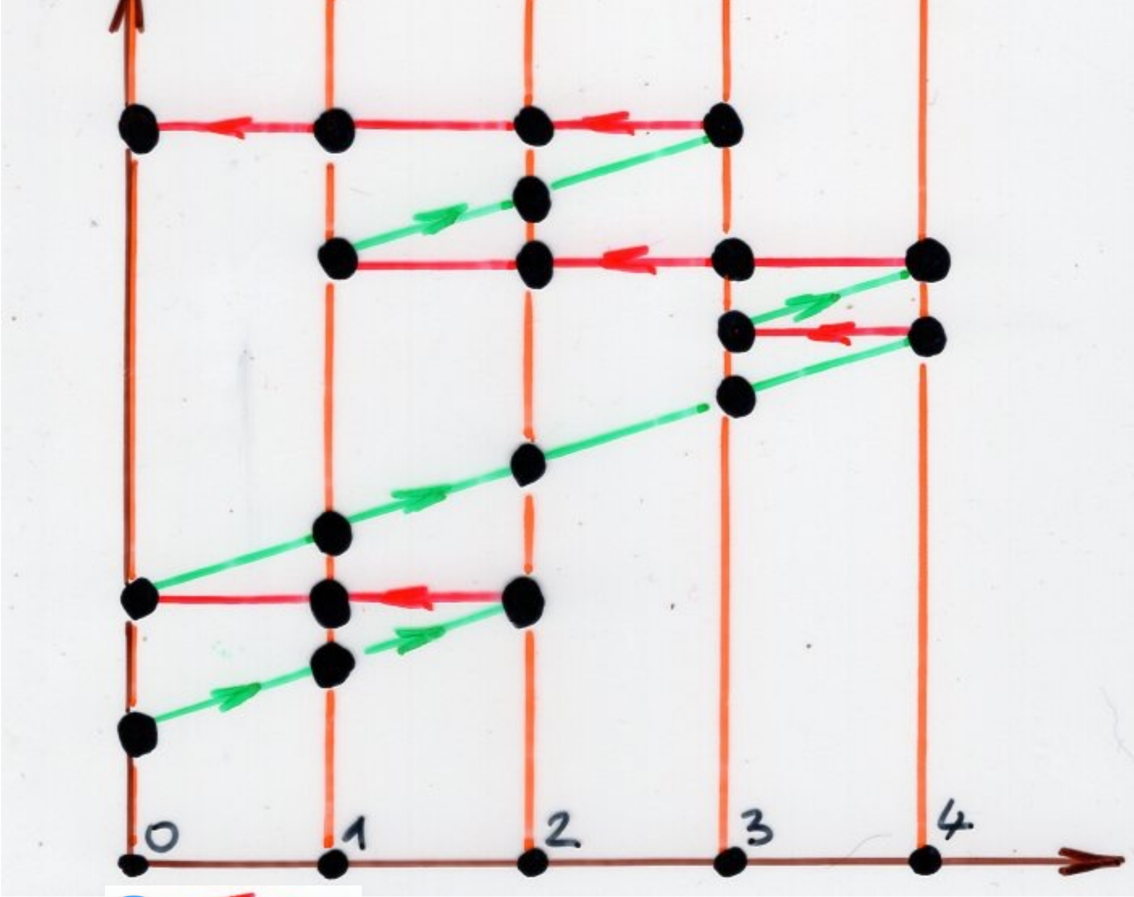


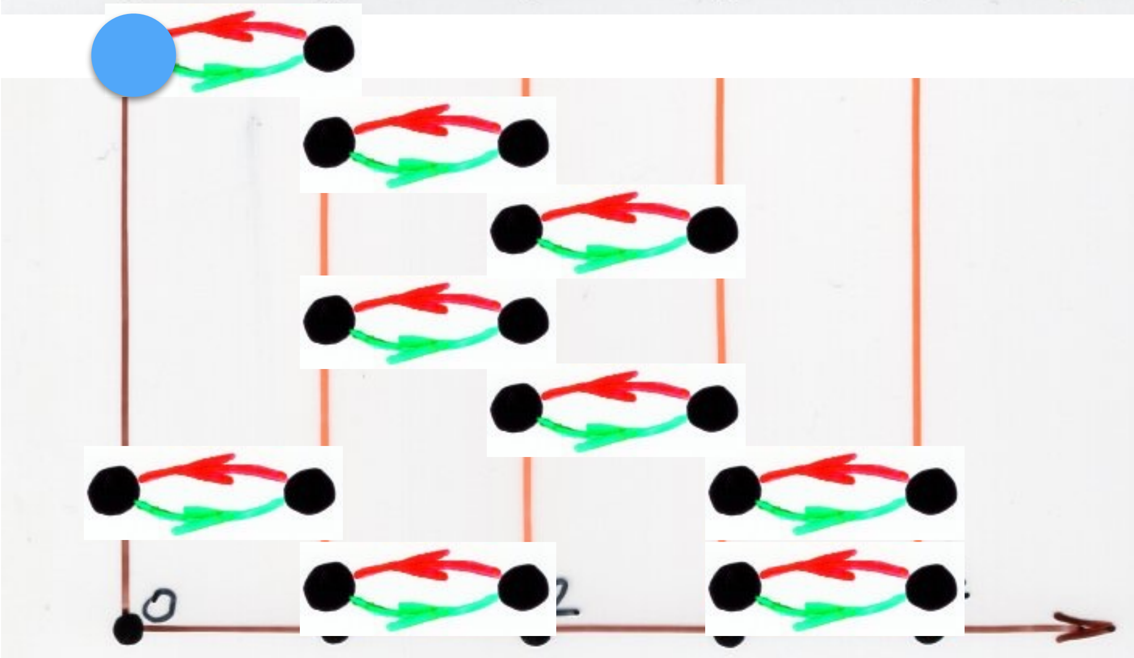
see the animation
on the video

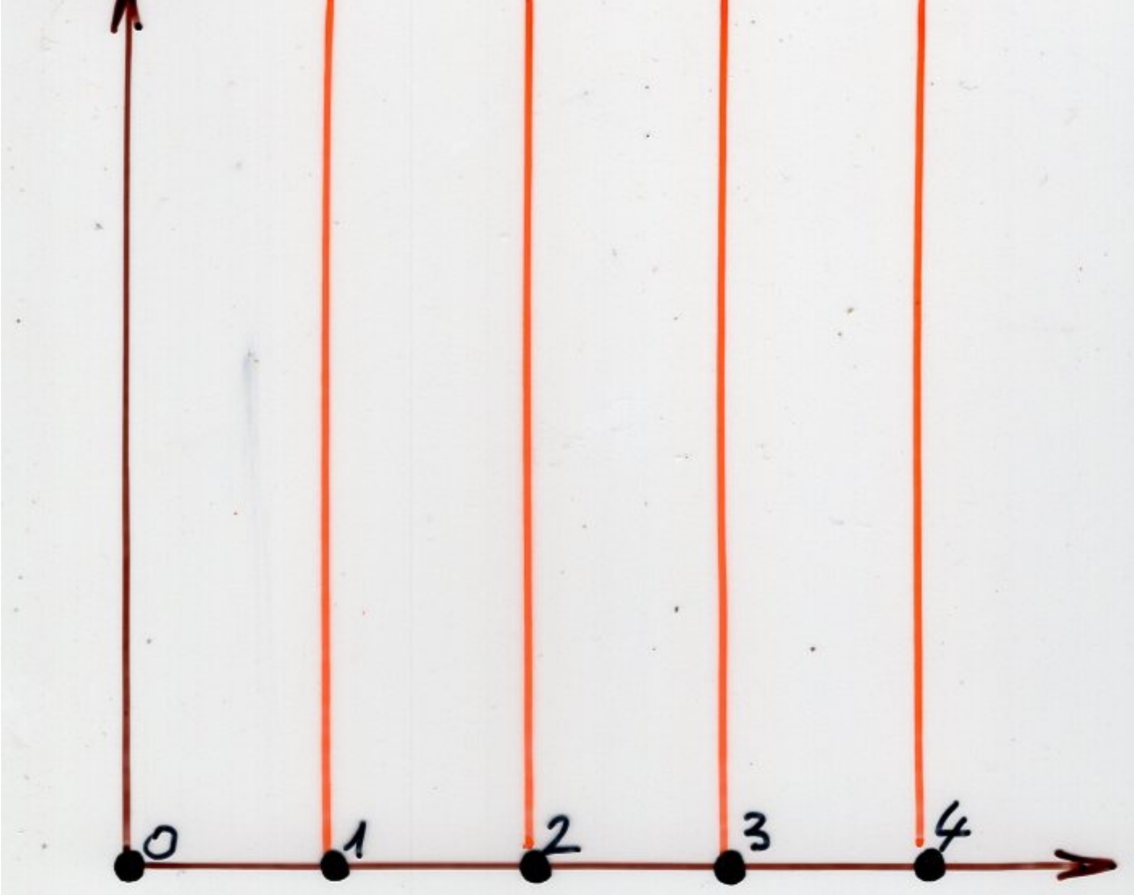
violin:
G. Duchamp

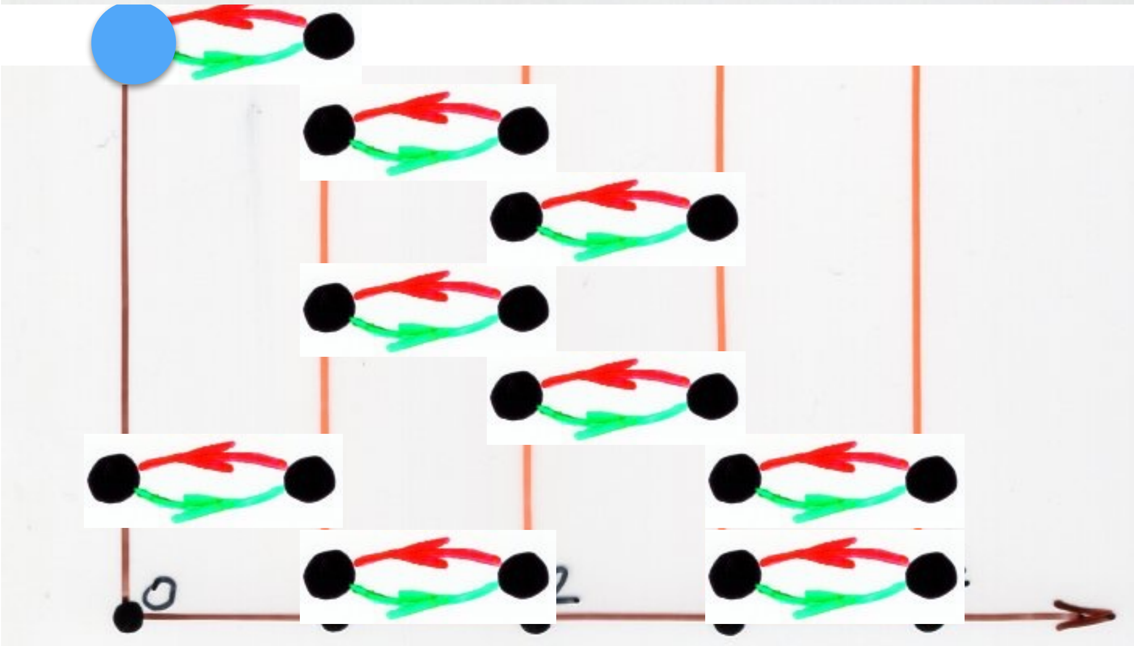

0
1
2
3
4

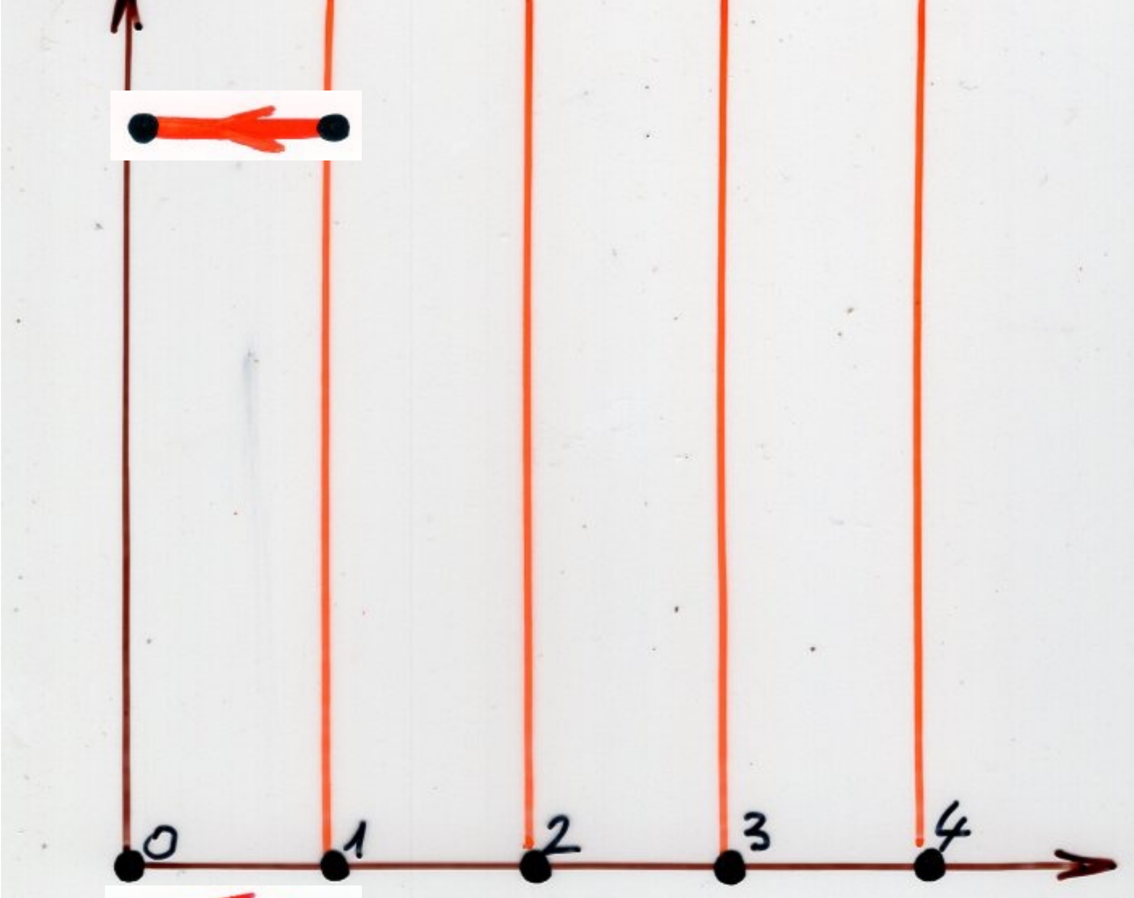

0

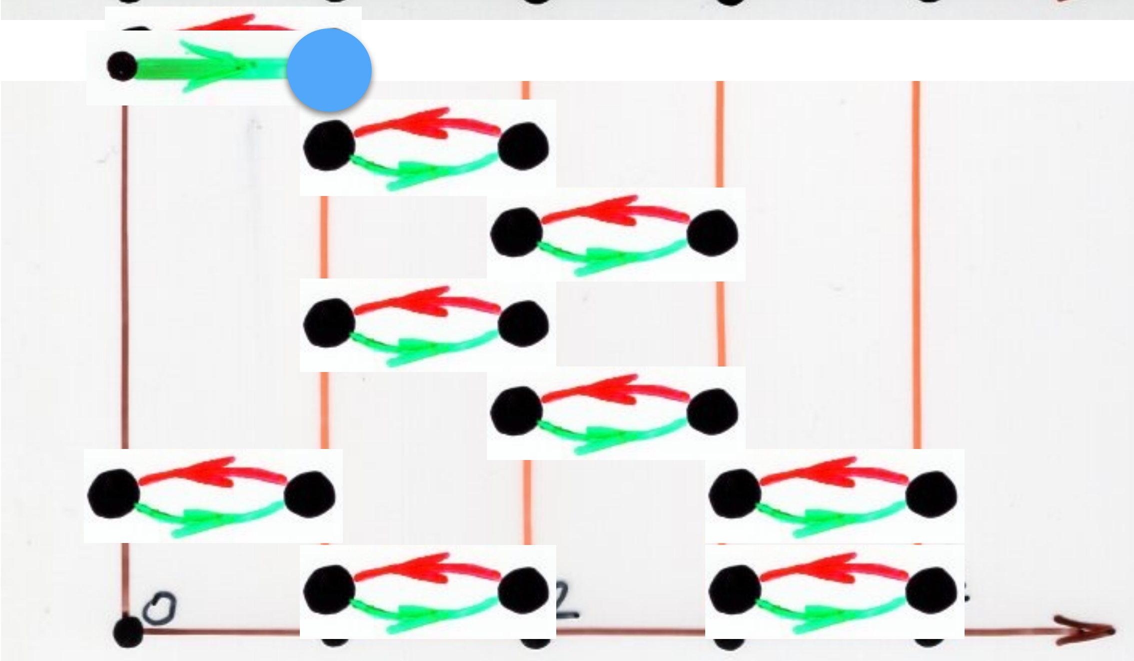

0
1
2
3
4

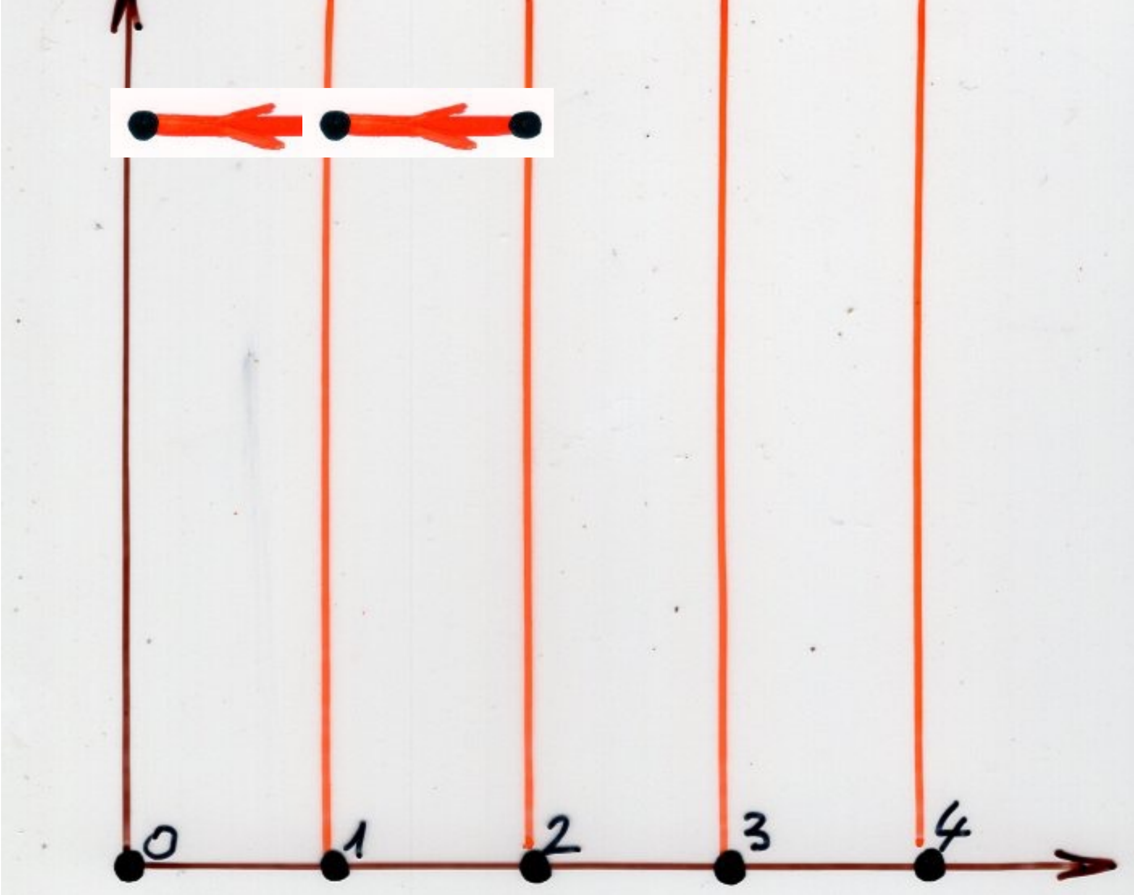

0


0
1
2
3
4

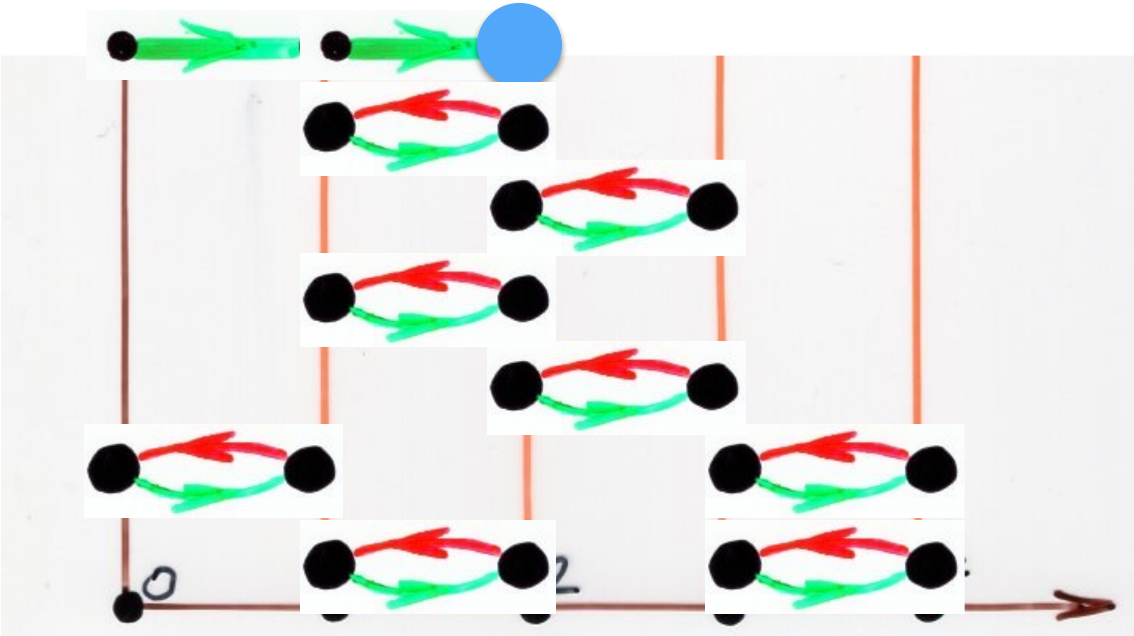

0


0
1
2
3
4

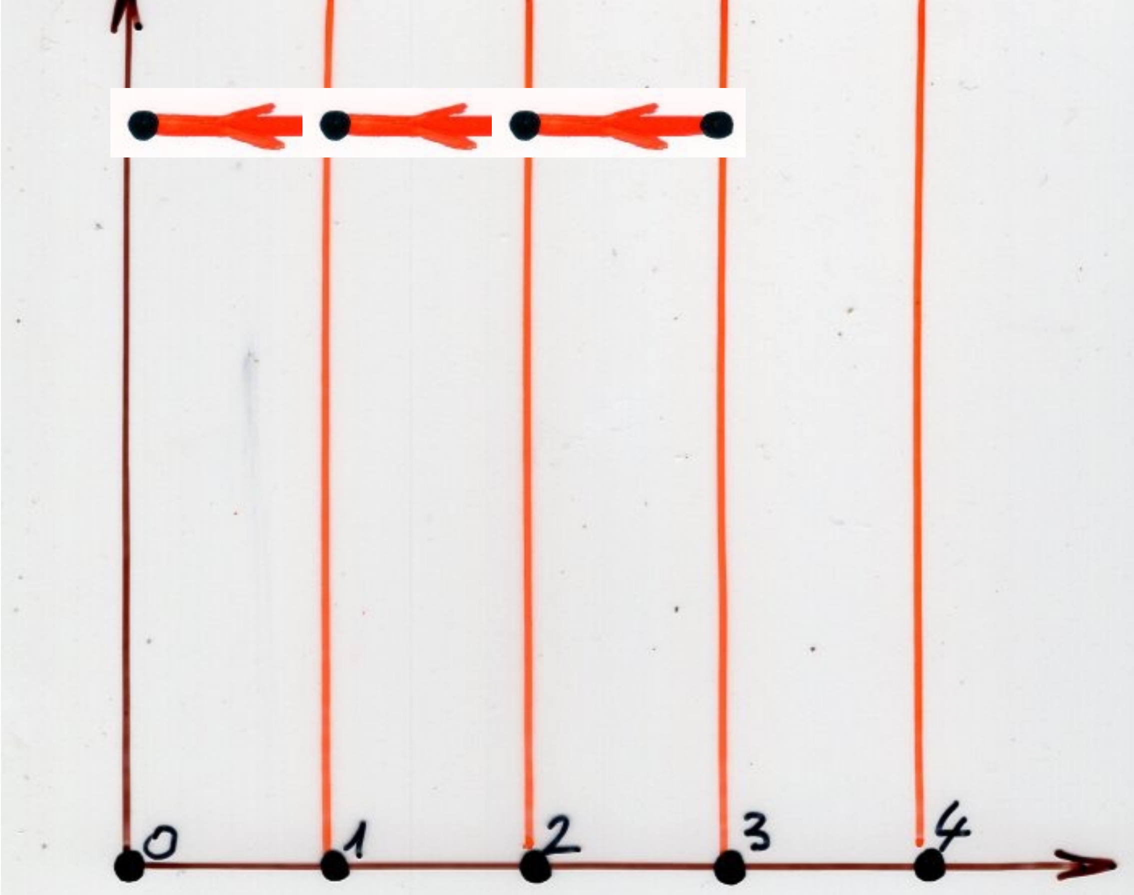

0

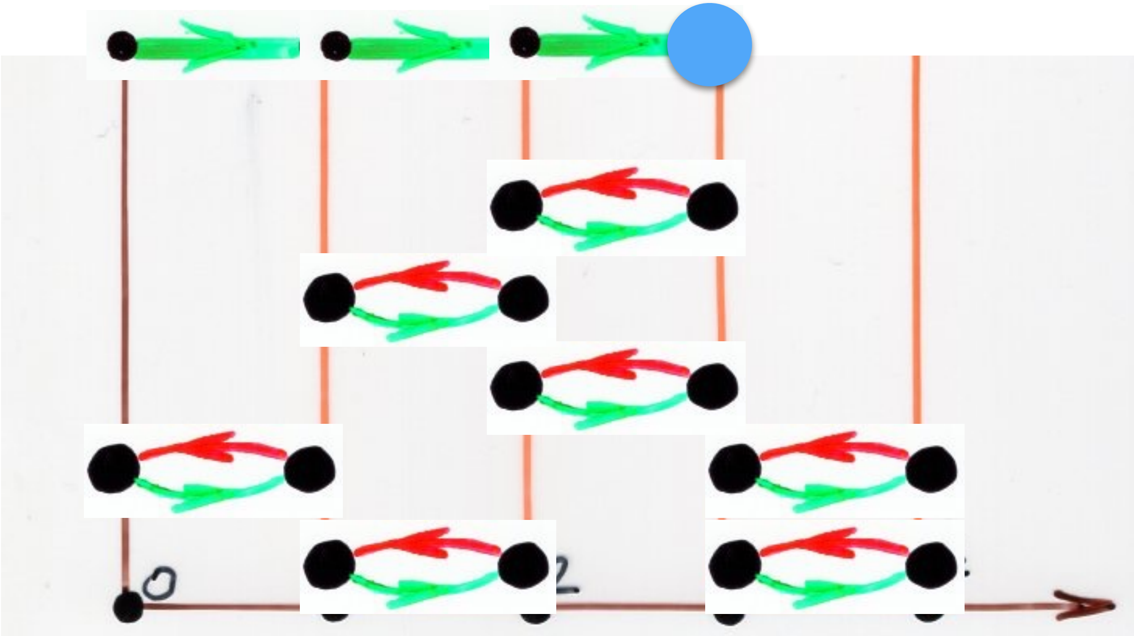


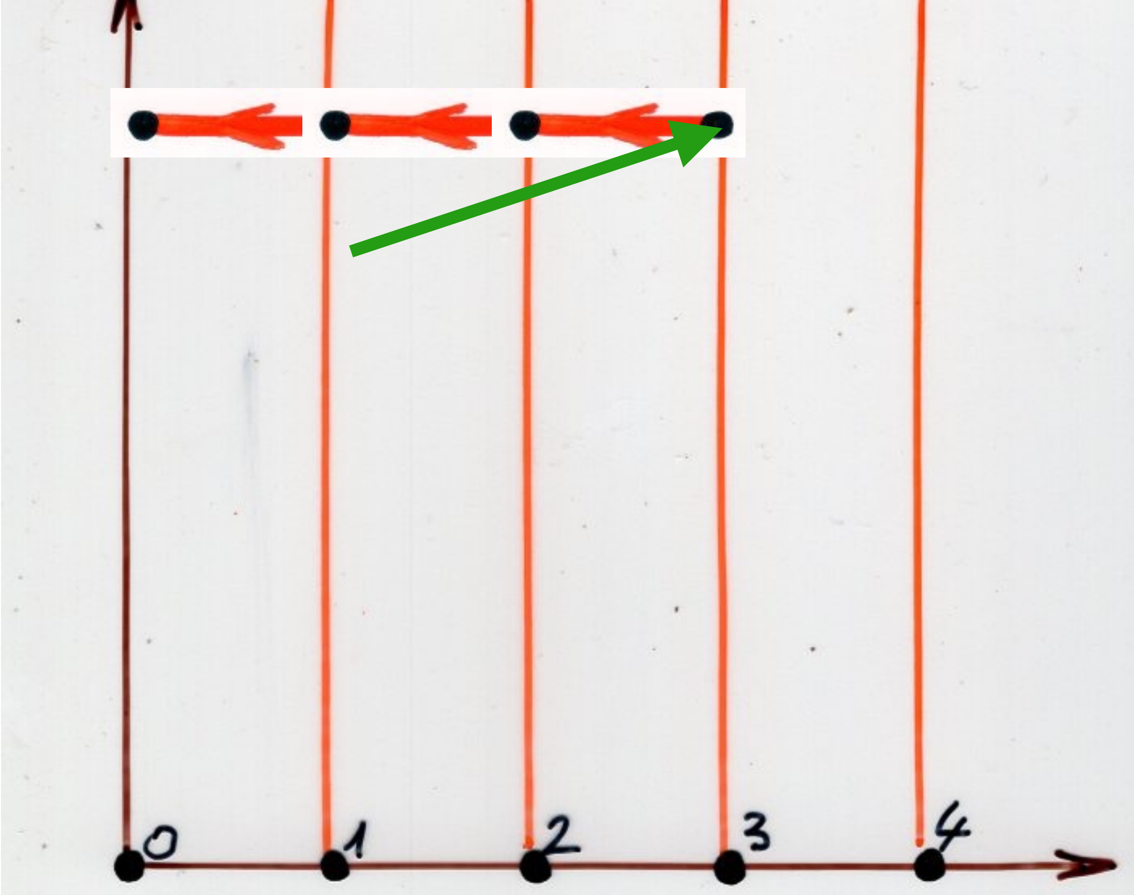
0
1
2
3
4

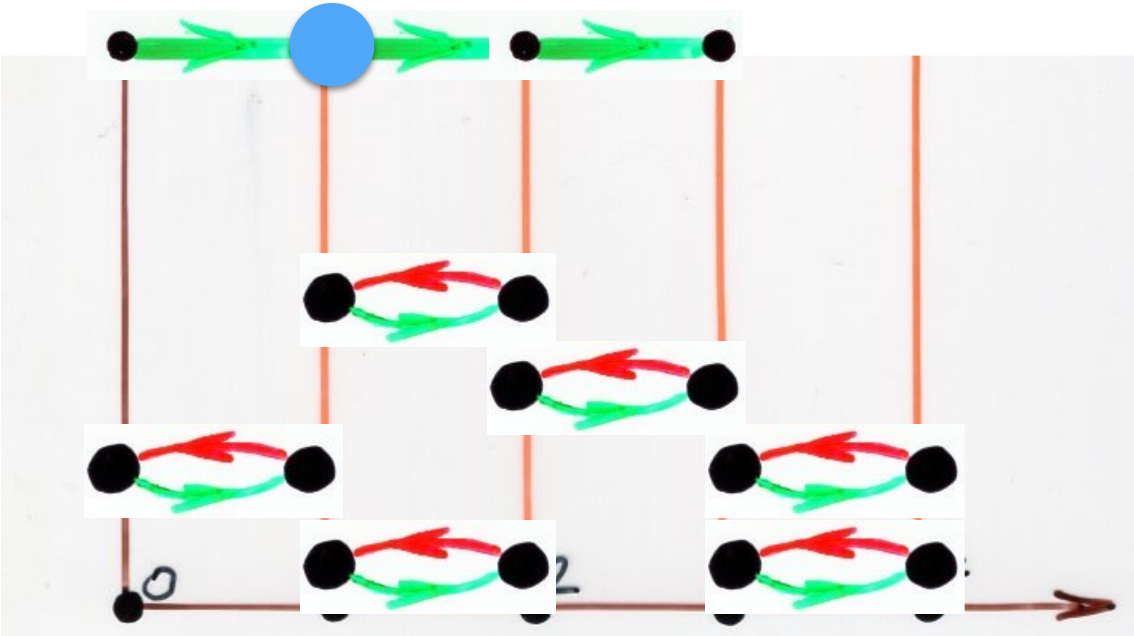


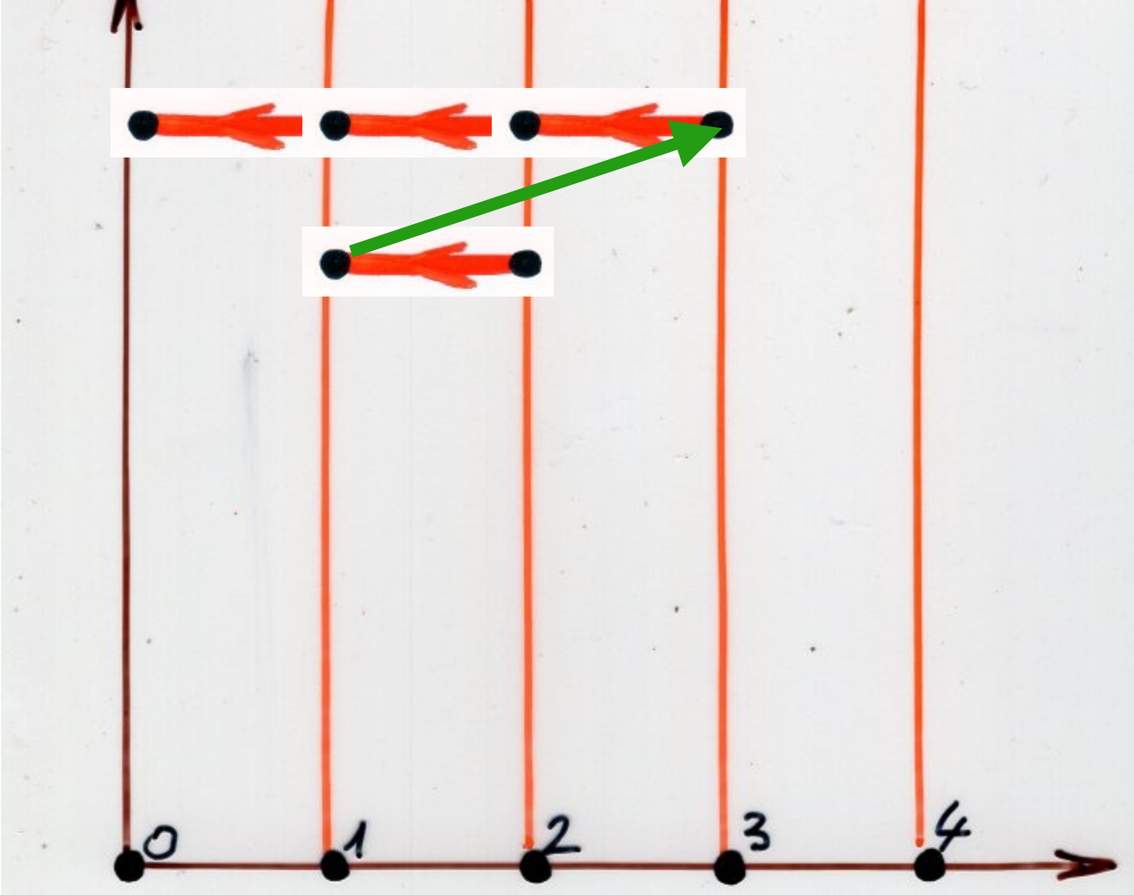
0

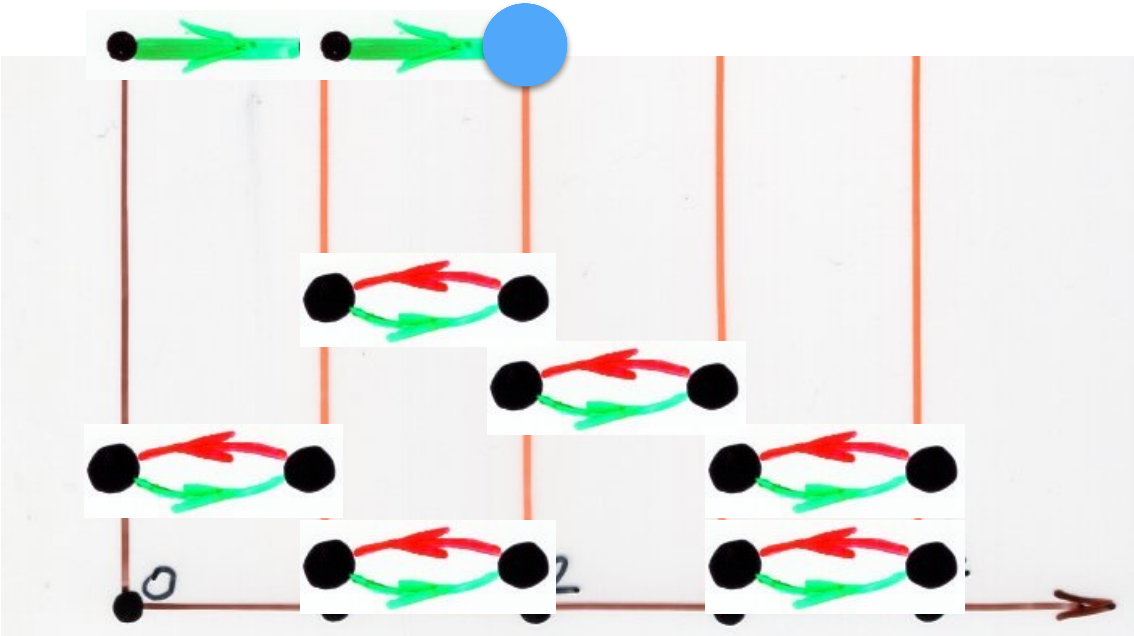

0
1
2
3
4

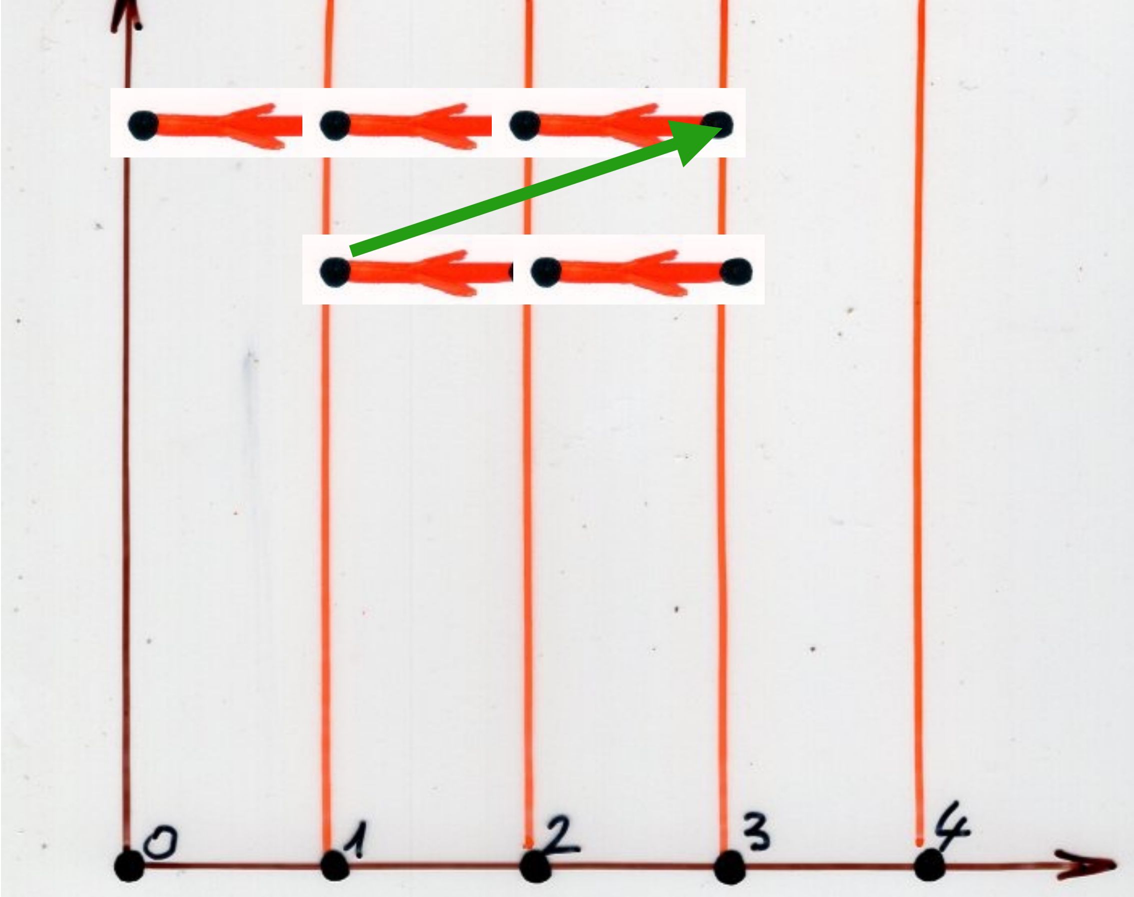

0

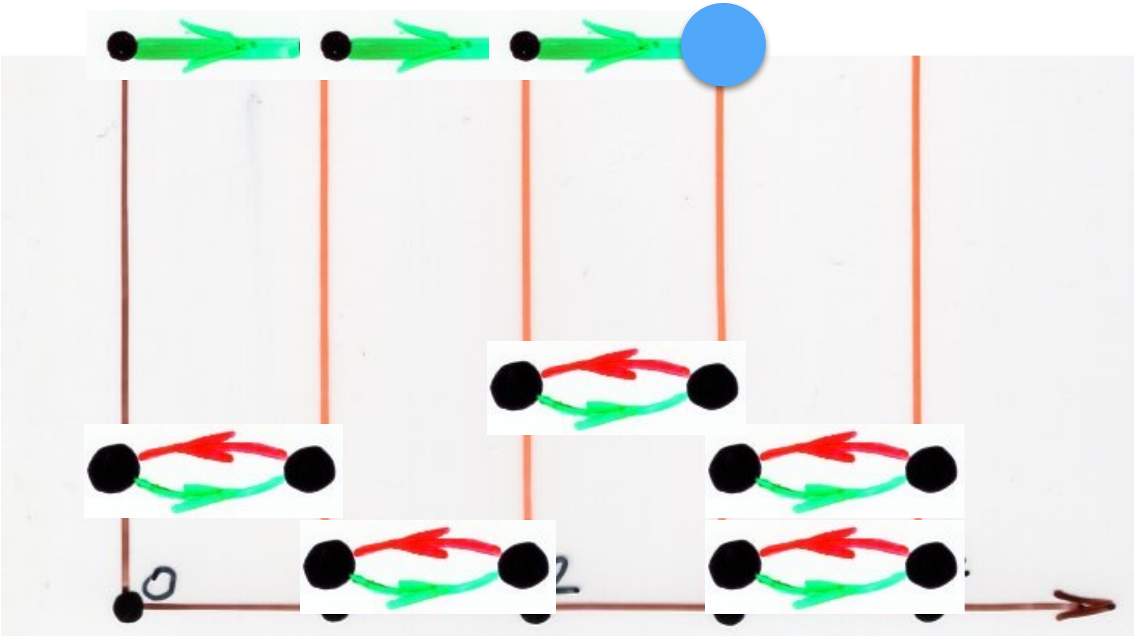

0
1
2
3
4

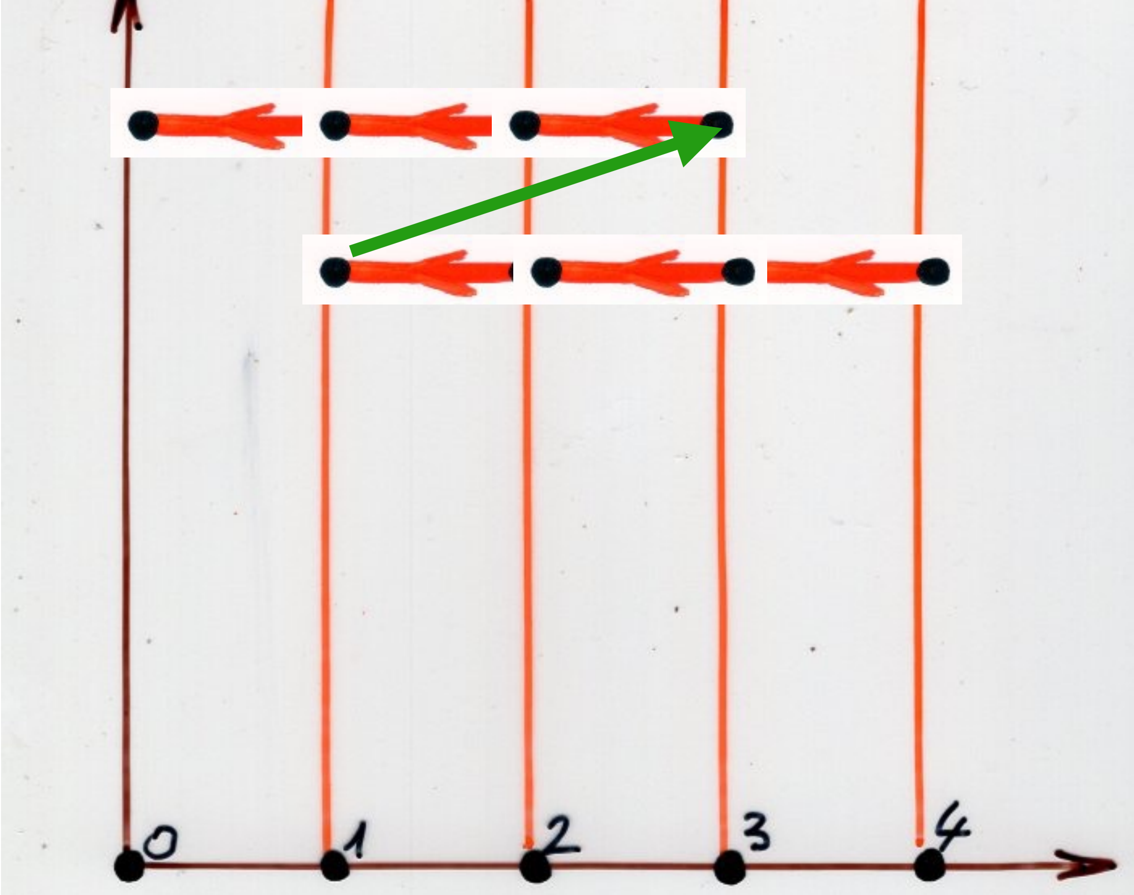

0
2

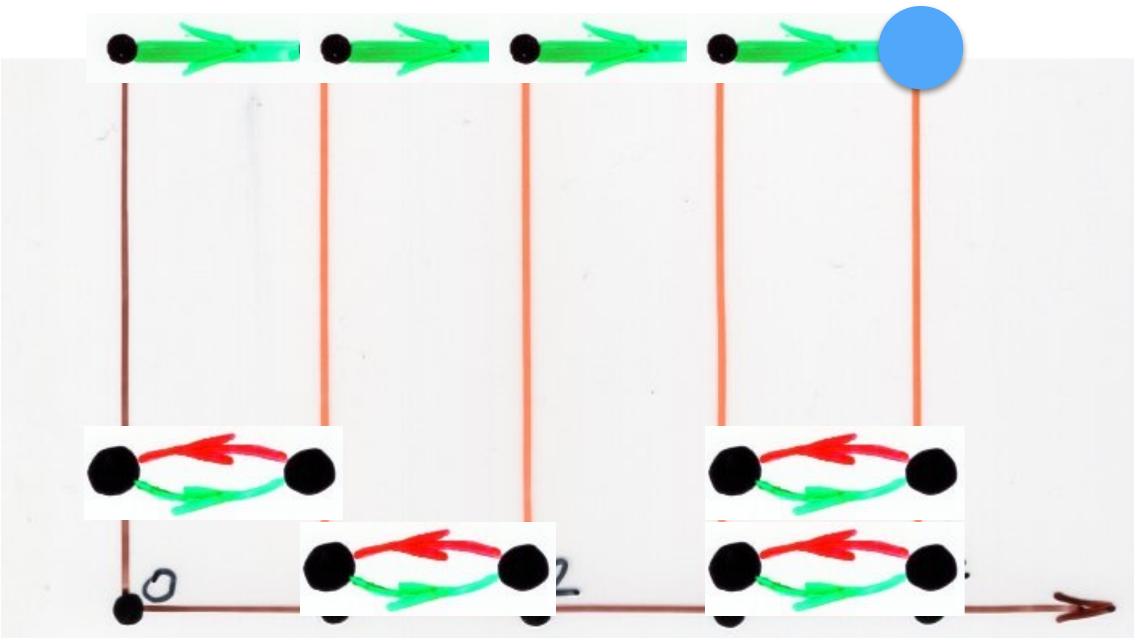

0
1
2
3
4

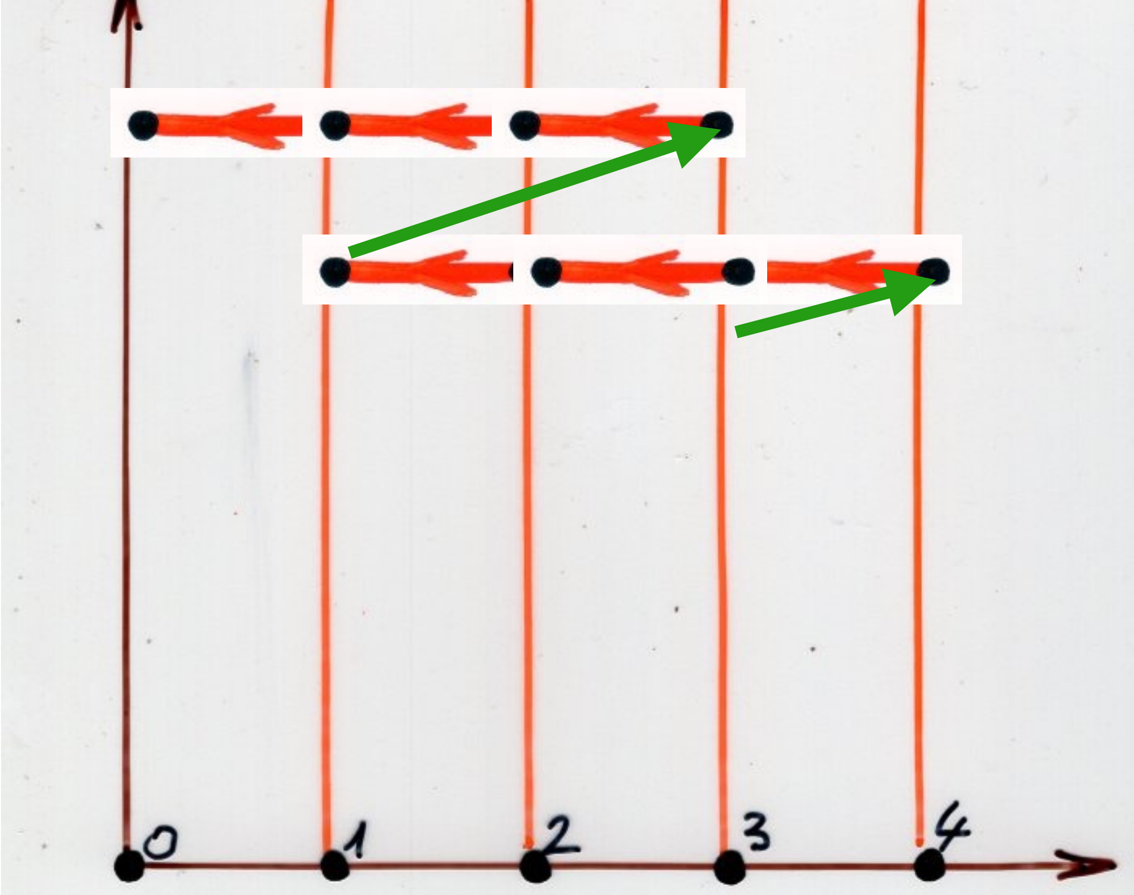

0

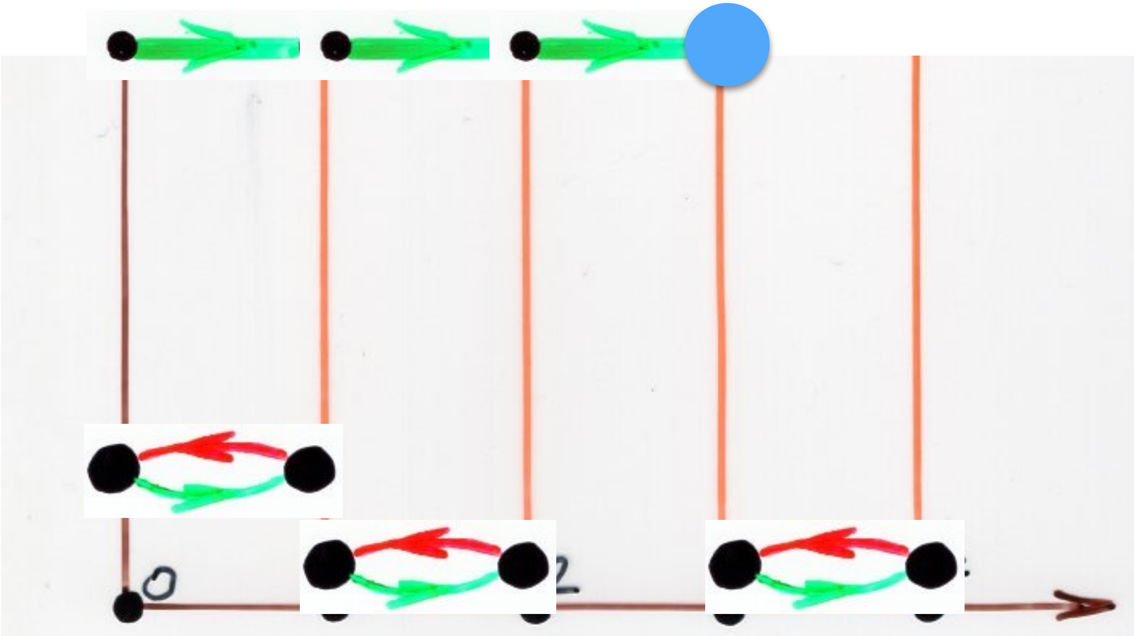

0
1
2
3
4

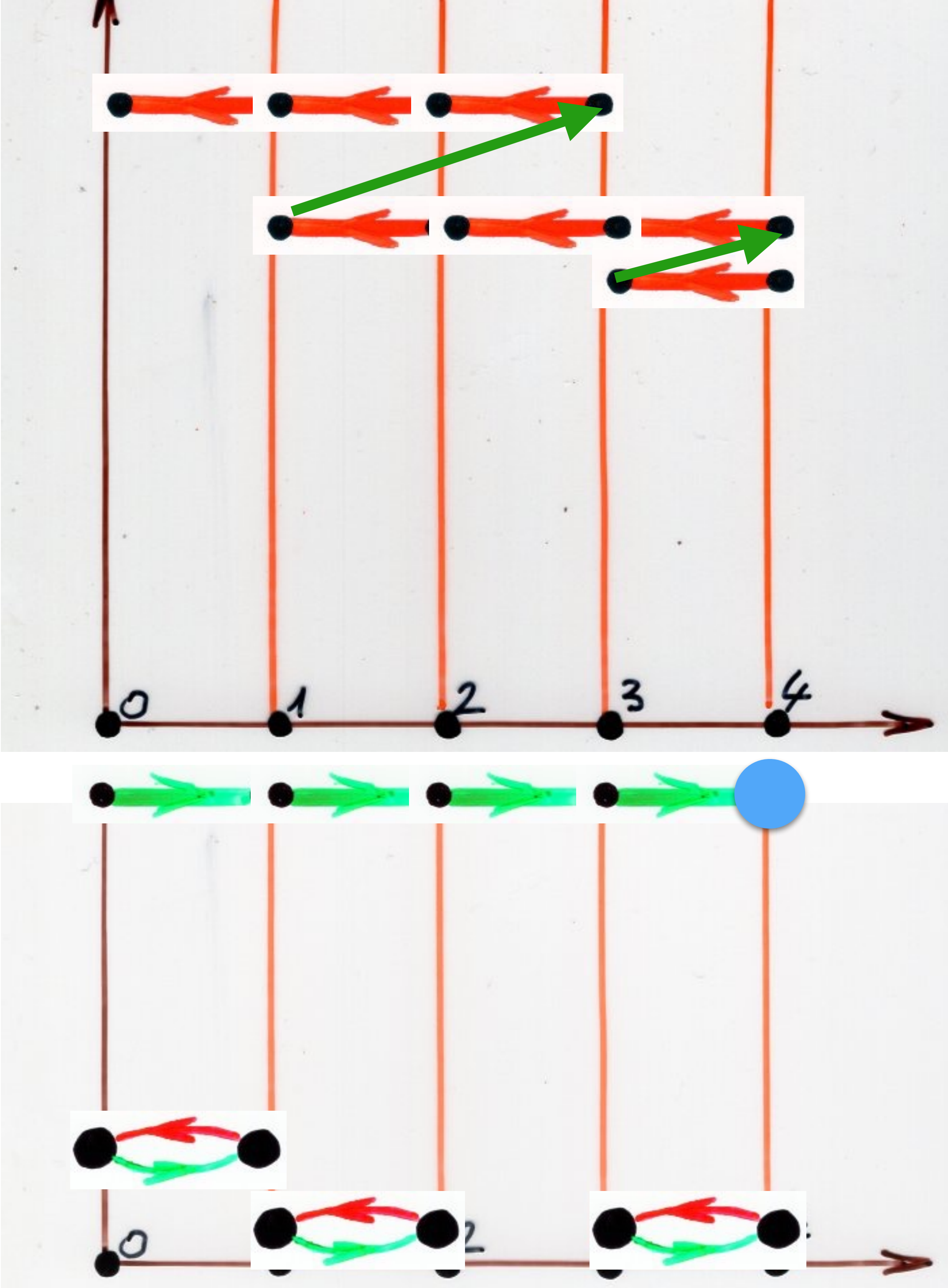

0

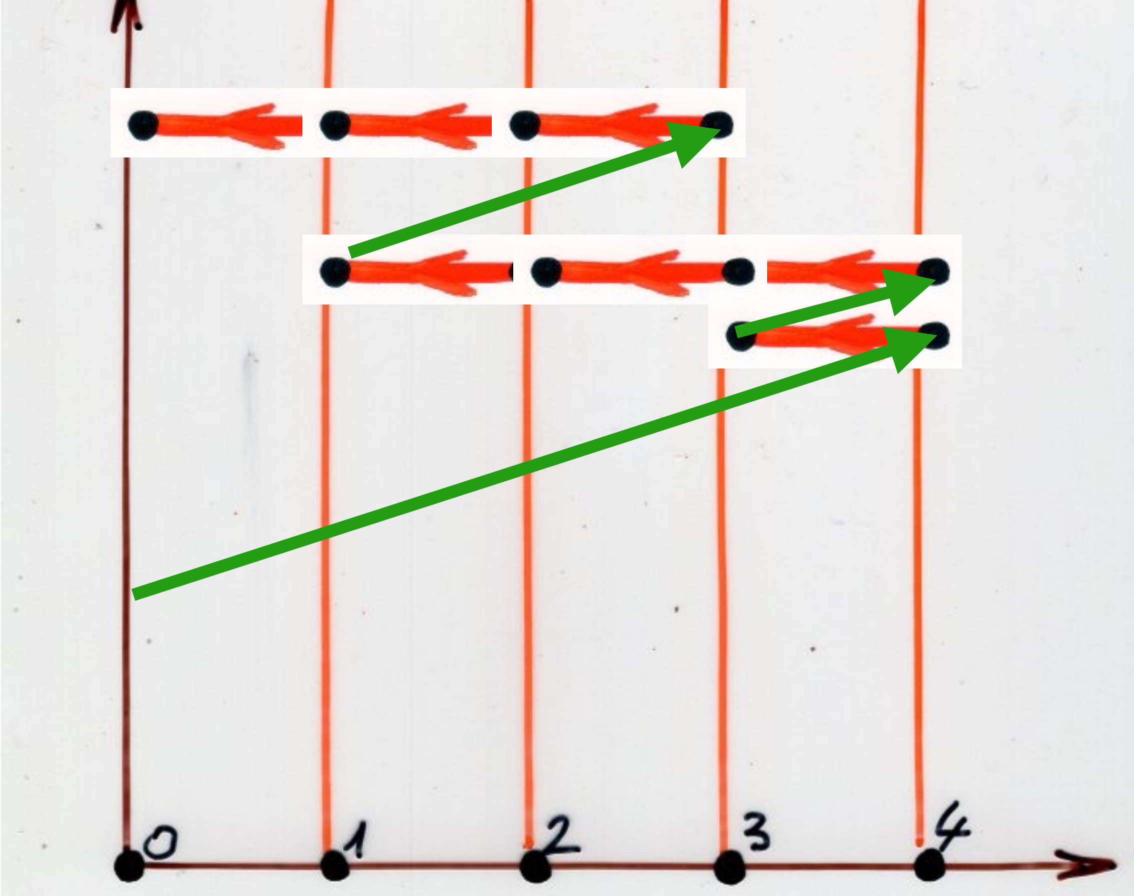

0
1
2
3
4


0

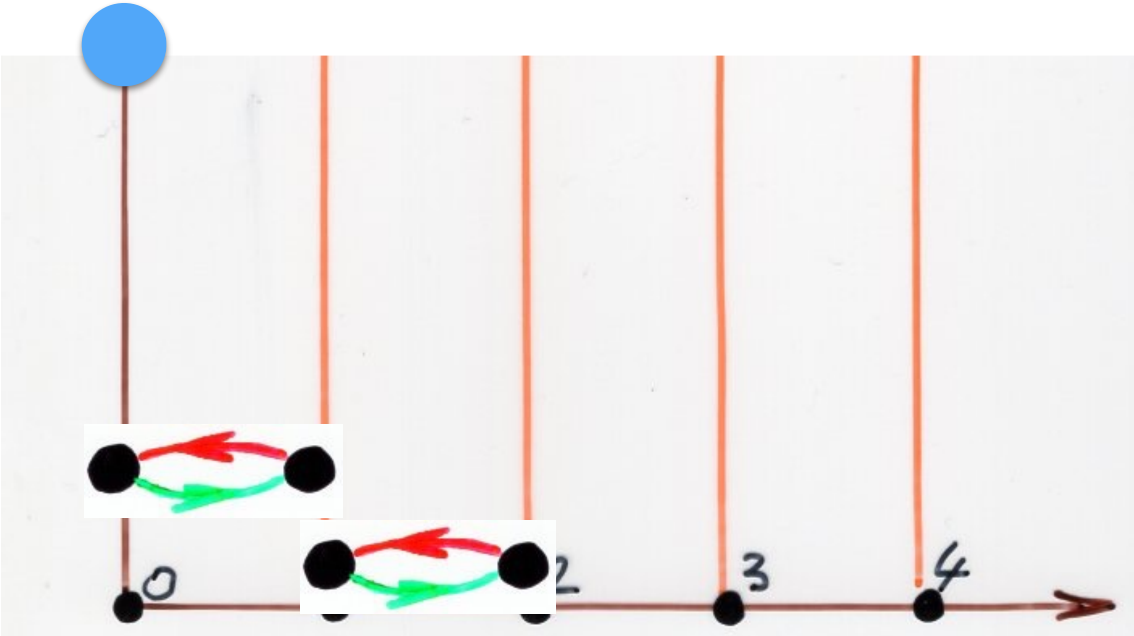

0
1
2
3
4


0

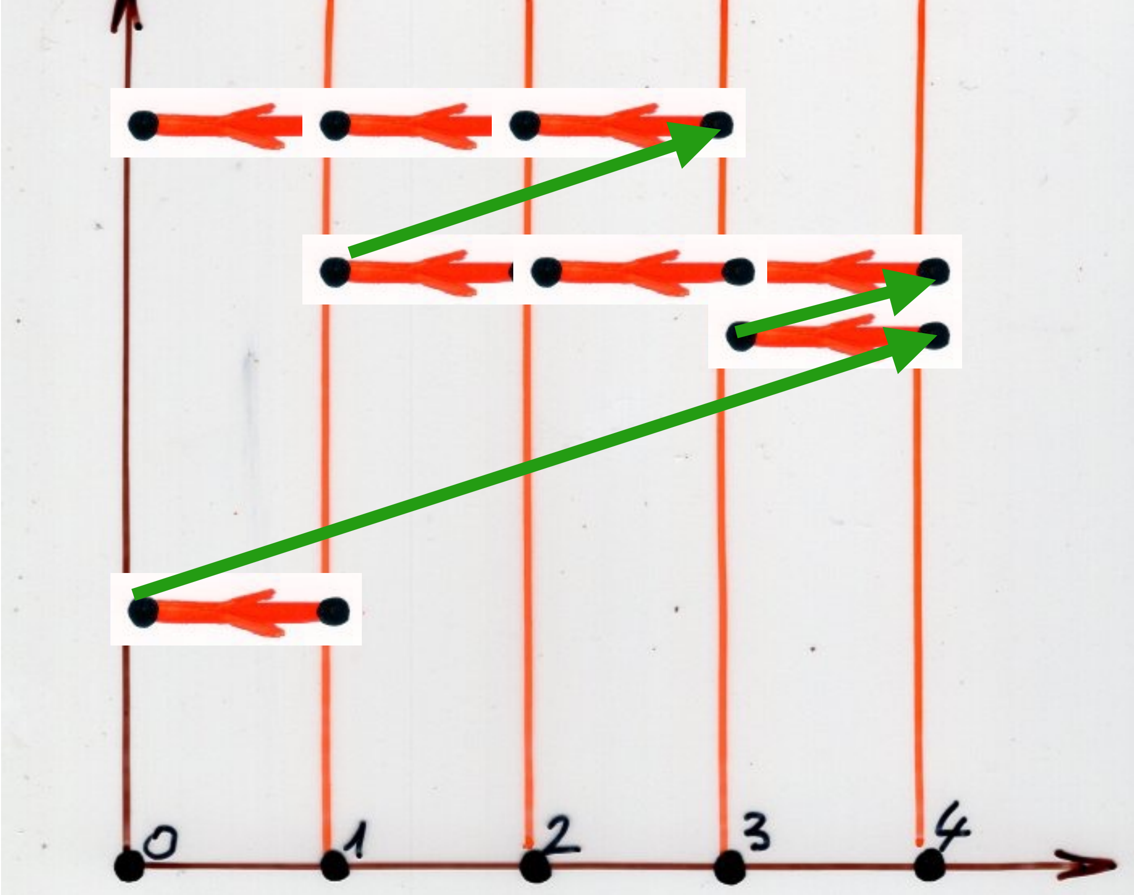

0
1
2
3
4
0
2

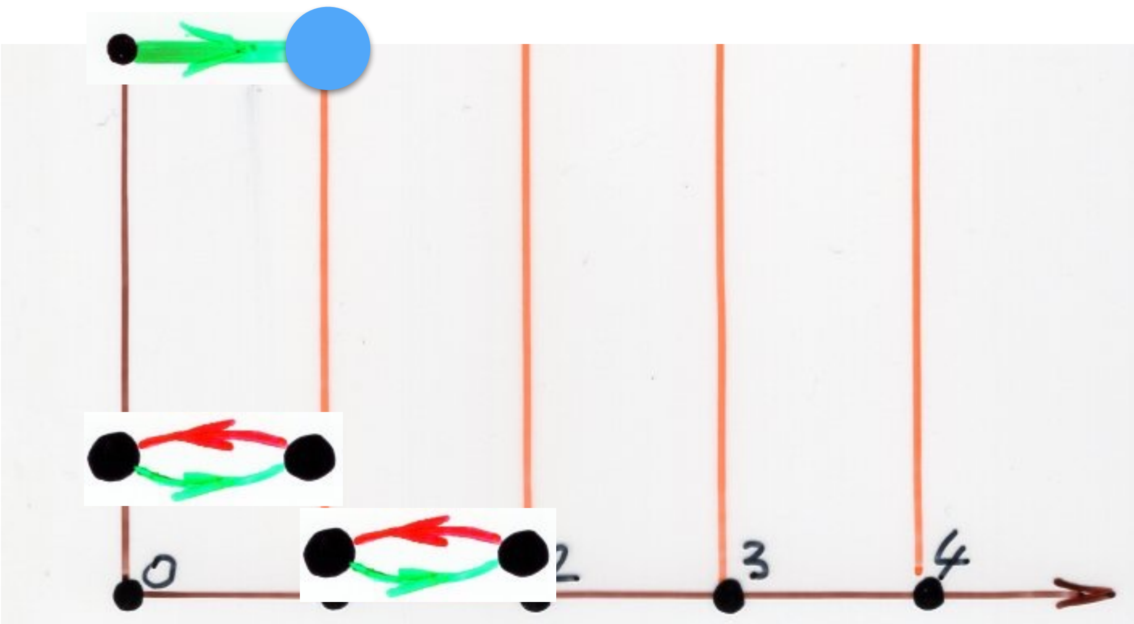


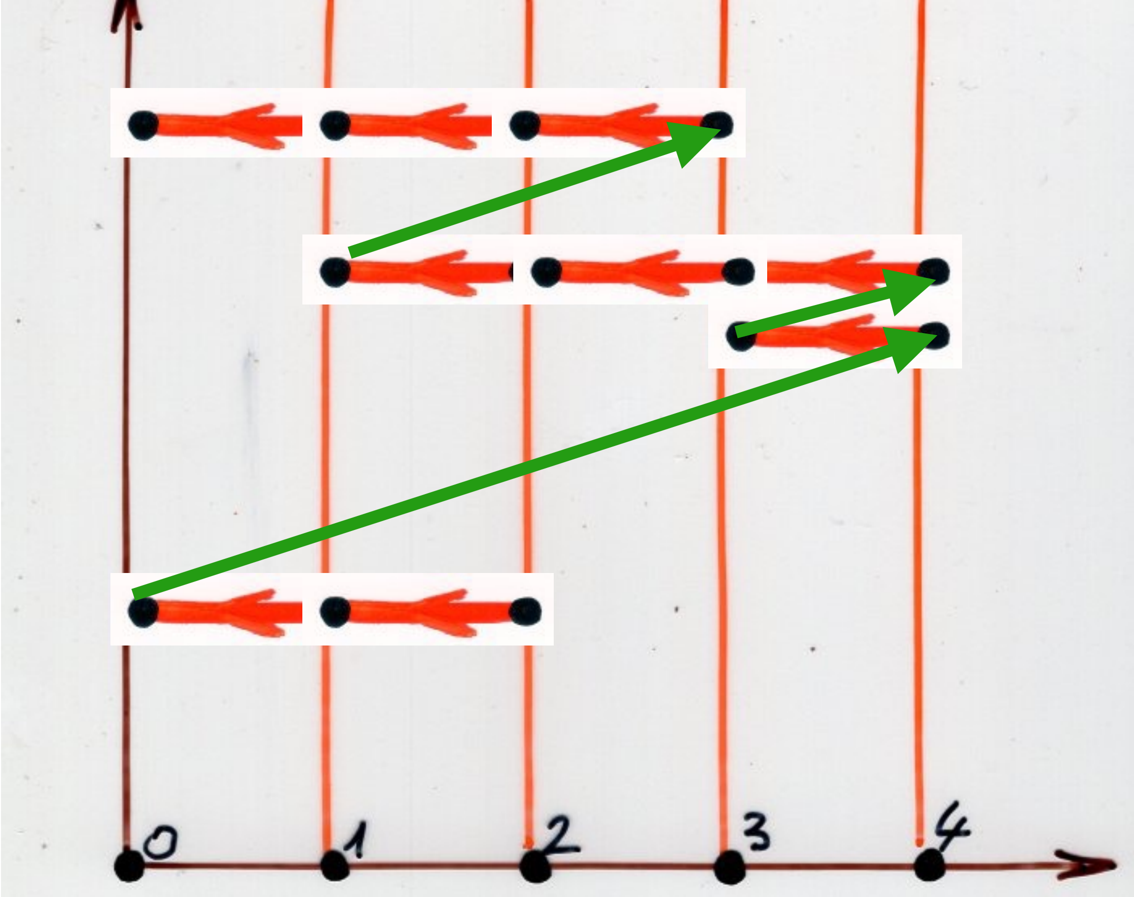
0
1
2
3
4

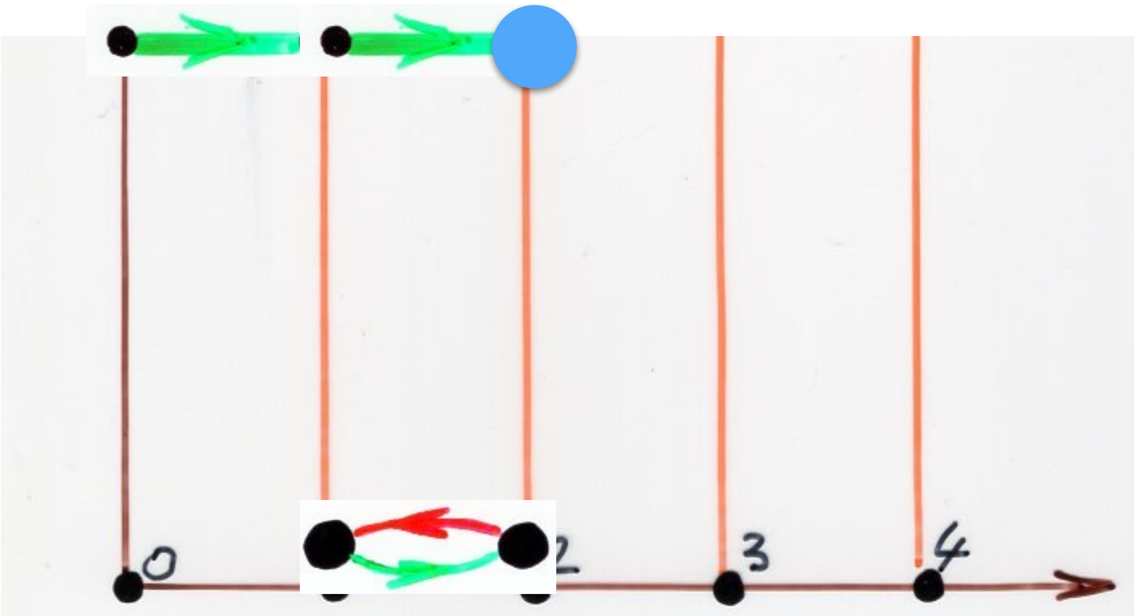


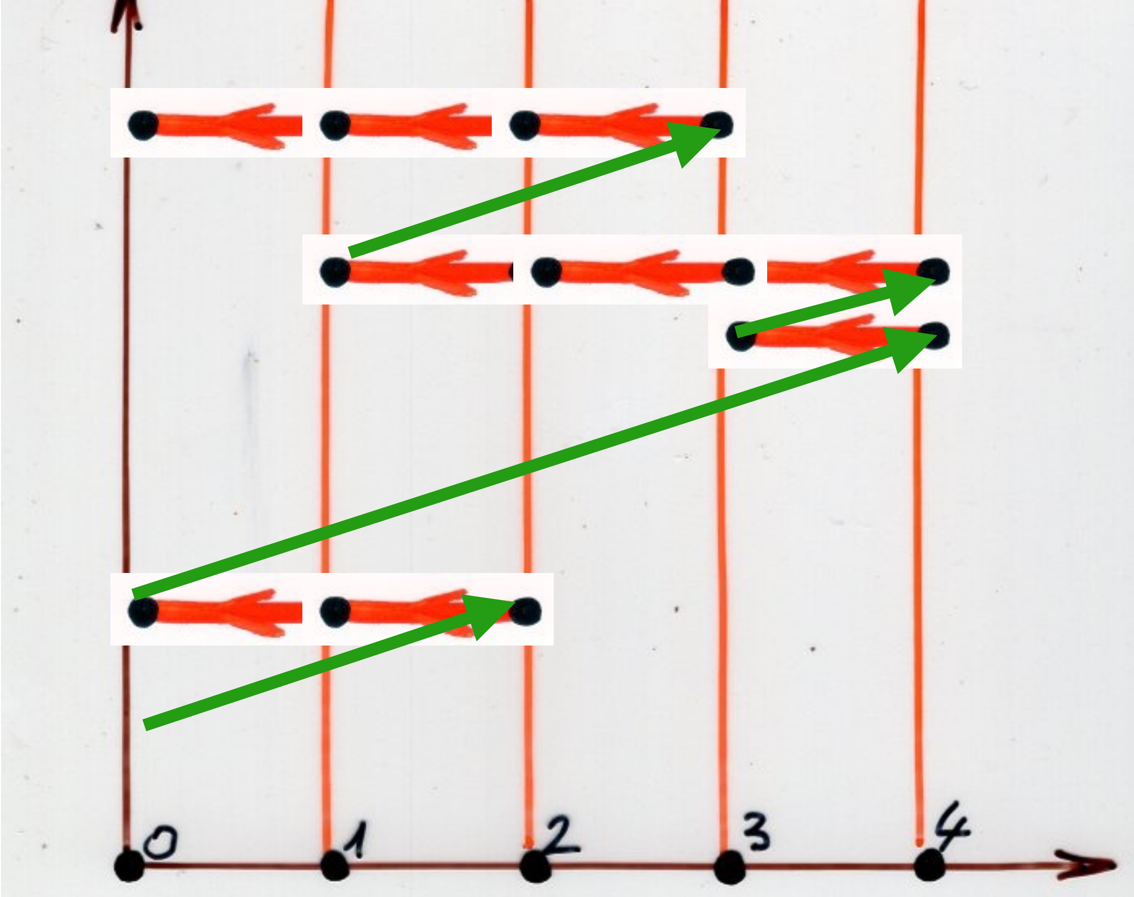
0
3
4

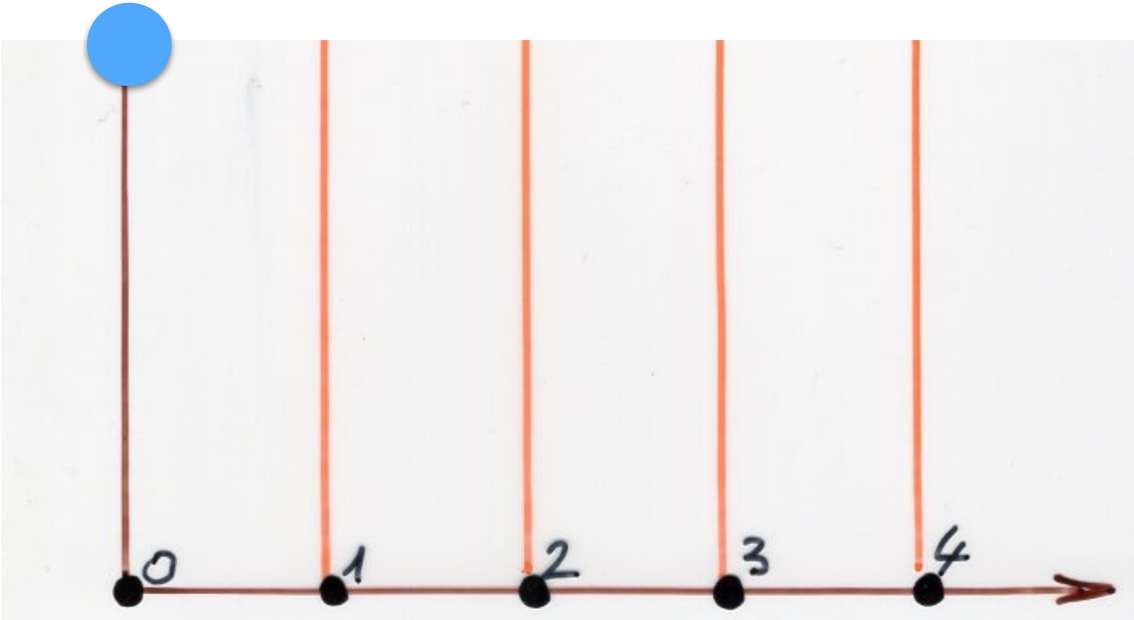

0
1
2
3
4

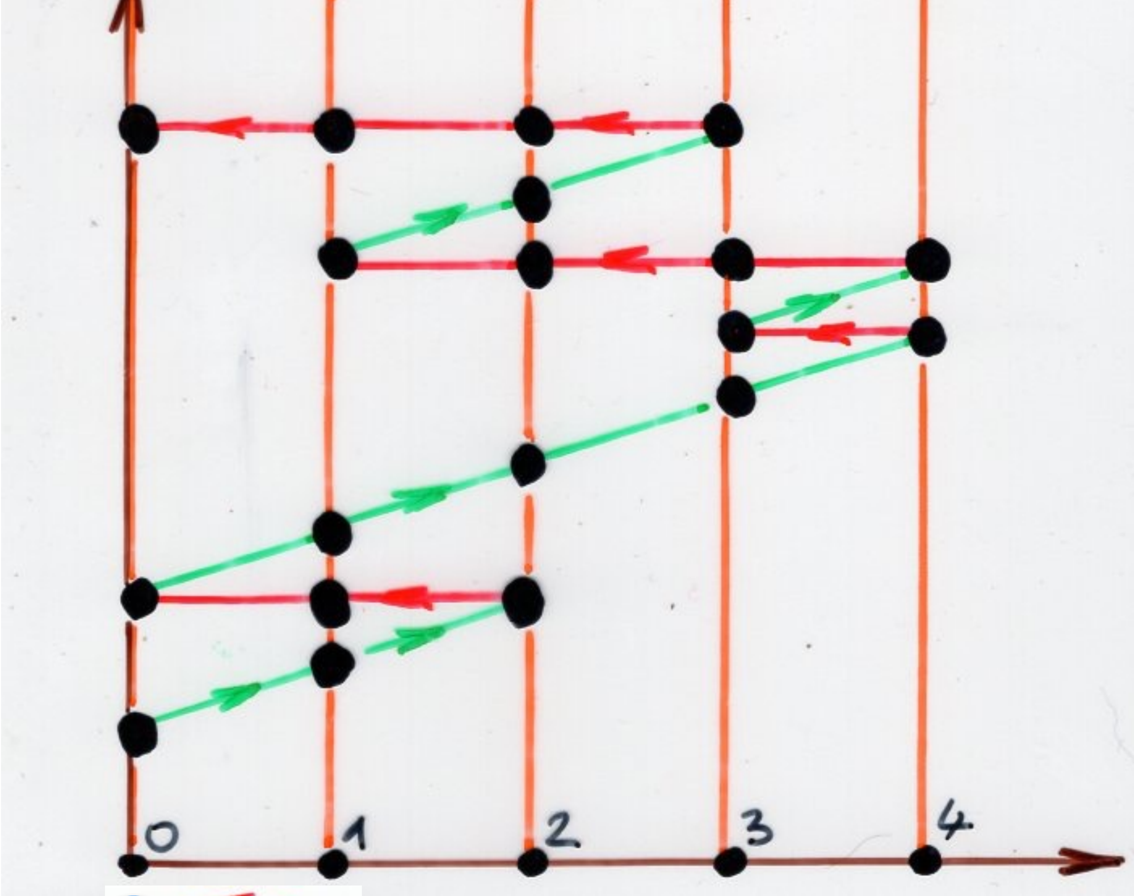

0
2
3
4

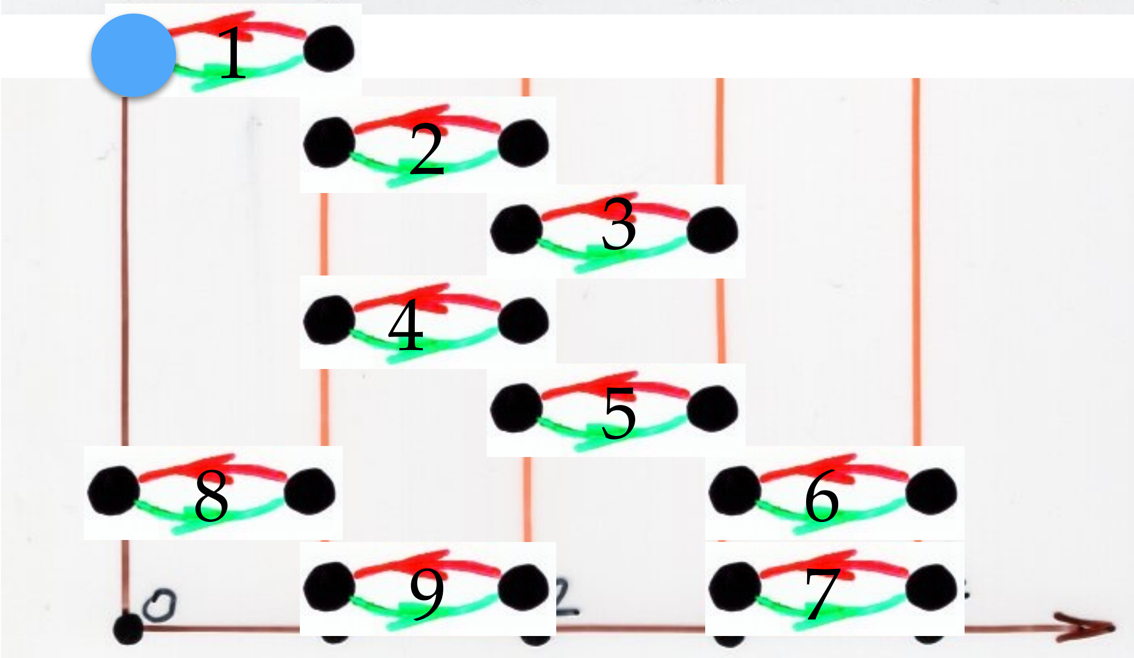

0
1
2
3
4

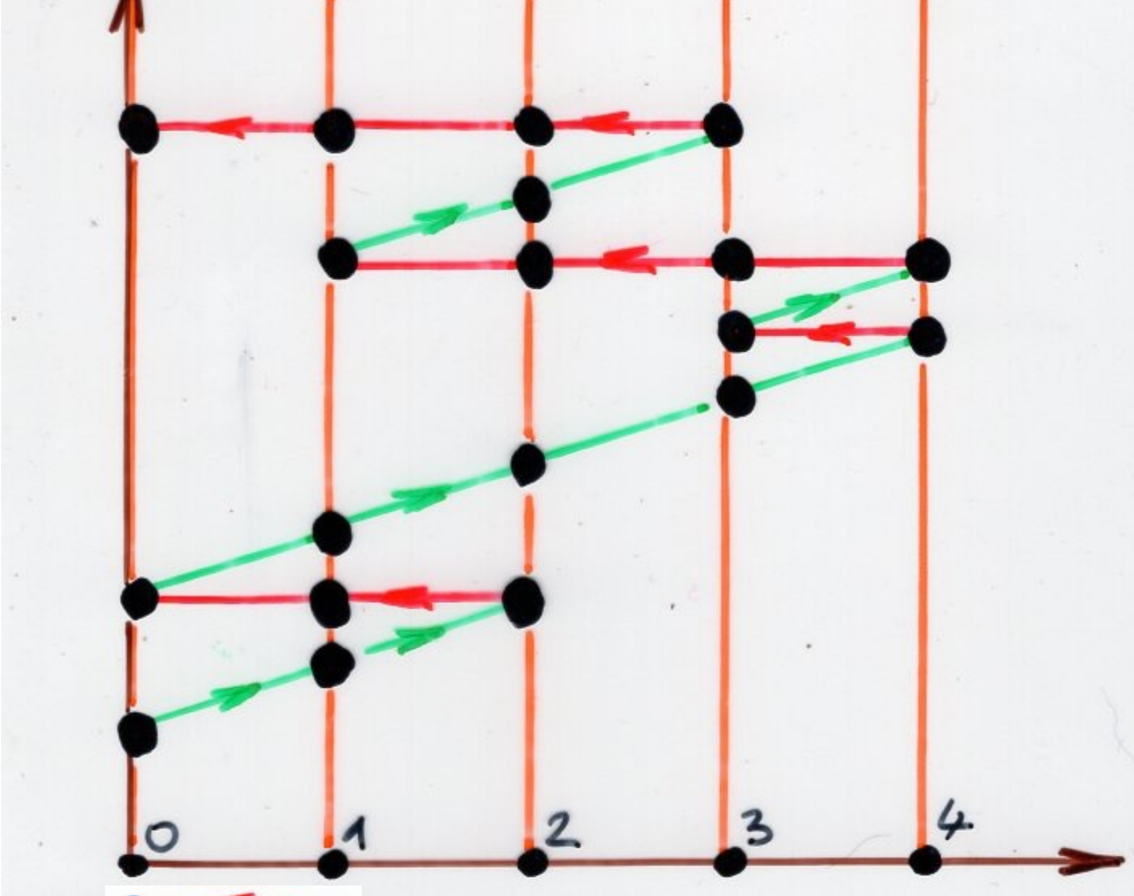

0
2
3
4

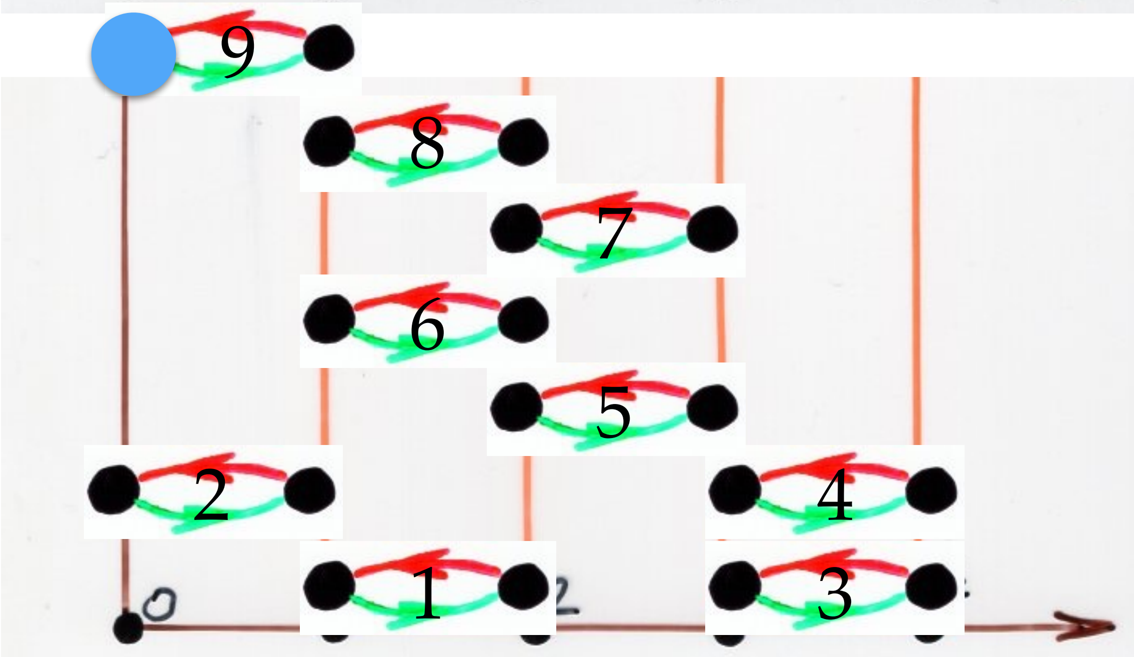

0
1
2
3
4


0
1
2
3
4


0
1
2
3
4


1
2
3
4
5
8
6
9
7
0


0
1
2
3
4

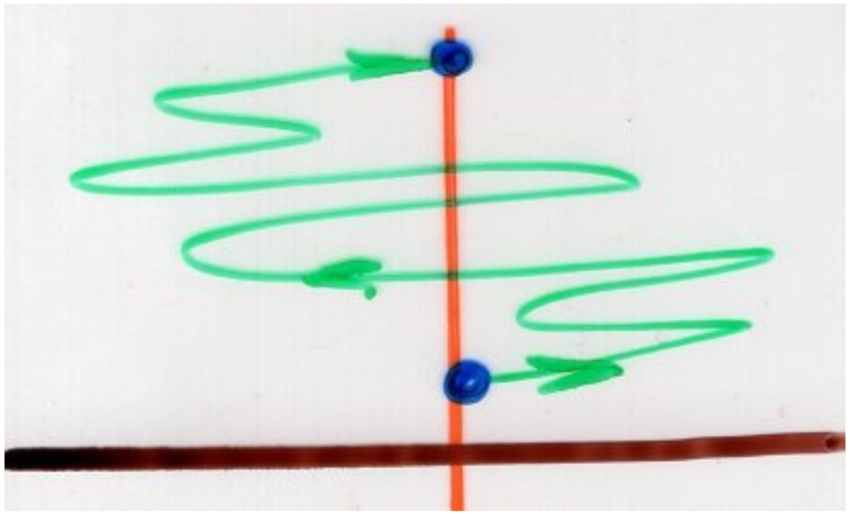

9
8
7
6
5
2
4
1
3
0

exercise 1

For bilateral Dyck paths explicit the general bijection and its reciprocal

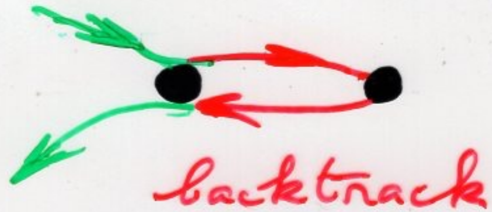
$\omega \longrightarrow$ pointed pyramid (of cycles of length 2)

(or pyramids of dimers on $\mathbb{Z}$)



Definition non-backtracking path

iff no pair of consecutive elementary step $(s_i, s_{i+1})\,(s_{i+1}, s_i)$



exercise 1 $\omega \to (\eta, E)$

(i) ω is non-backtracking

(ii) the heap E has no cycles of length 2

does (i) $\Rightarrow$ (ii) ?

(ii) $\Rightarrow$ (i) ?

__definition__ G graph, χ

ω path on G with $\omega \longrightarrow (\eta, E)$.

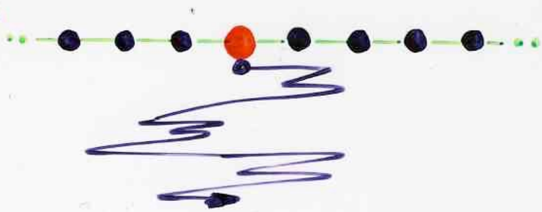
ω is tree-like iff the heap E contains only cycles of length 2.

Godsil (1981)

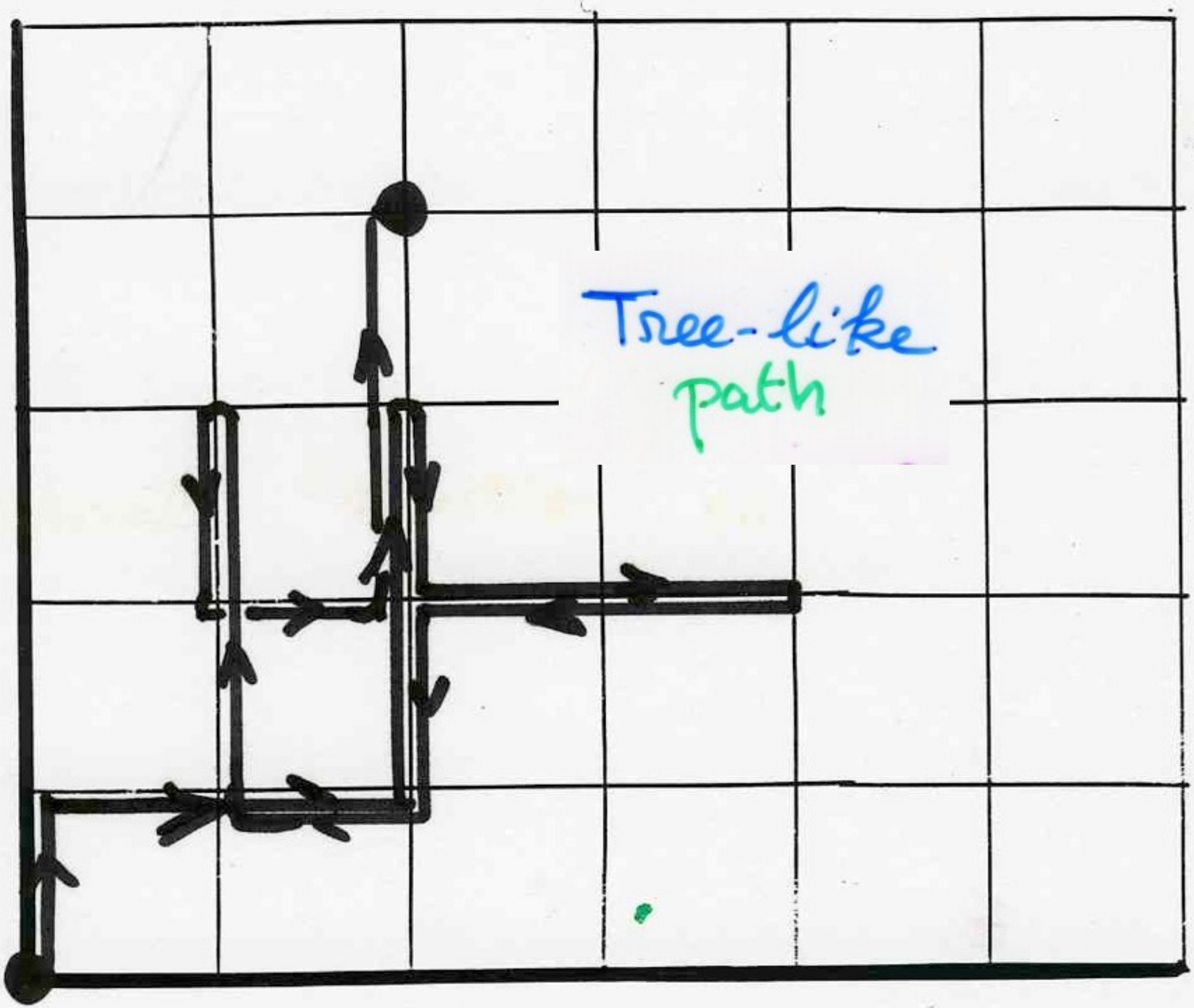
__Particular cases.__

- **Dyck** path

- bilateral **Dyck** paths



paths on a tree


Tree-like
path

<u>definition</u> G graph, X

ω path on G with $\omega \longrightarrow (\eta, E)$.

ω is tree-like iff the heap E contains only cycles of length 2.

<u>exercise</u> **3** G graph, s vertex of G

Construct a tree T such that the tree-like paths on G starting at s are in bijection (preserving the length) with the paths on T starting at the root of T

complements

LERW

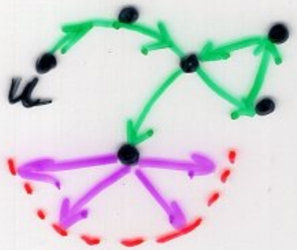
"Loop-erased random walks"

LERW

Loop-erased random walk

Lawler (1980)

ω random path on X



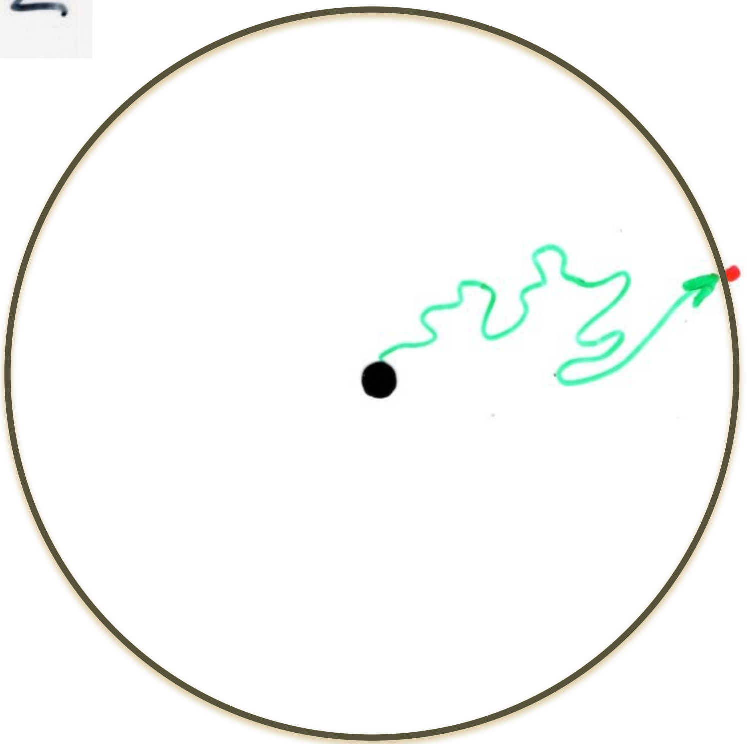
Markov chain

$$\omega \xleftrightarrow{X} (\eta, E)$$

path ω $u \rightsquigarrow v$ $\longrightarrow$ self-avoiding path η $u \rightsquigarrow v$

probability law on η

$D = 2$



LERW $\longrightarrow$ SLE$_2$

Schramm-Loewner
evolution

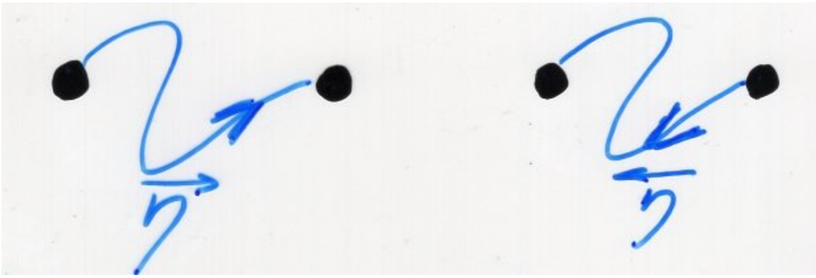
length of ω $\quad n^{5/4}$

scaling limit
of random planar curves

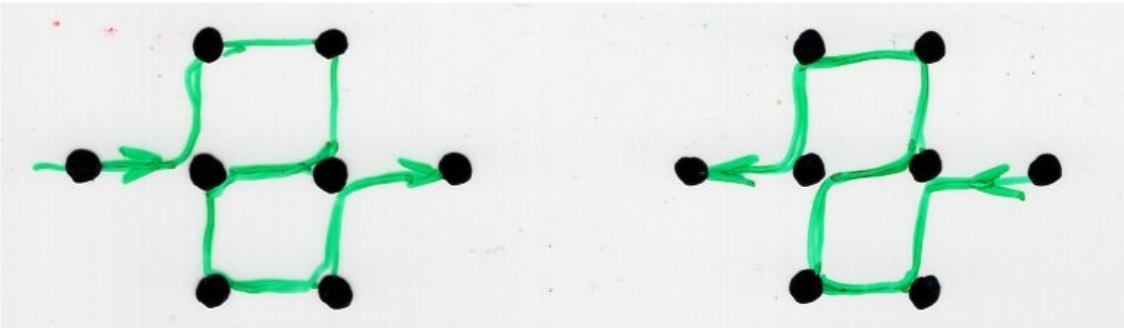
- LERW loop-erased random walk
- ASM abelian sandpile model
- dimer model

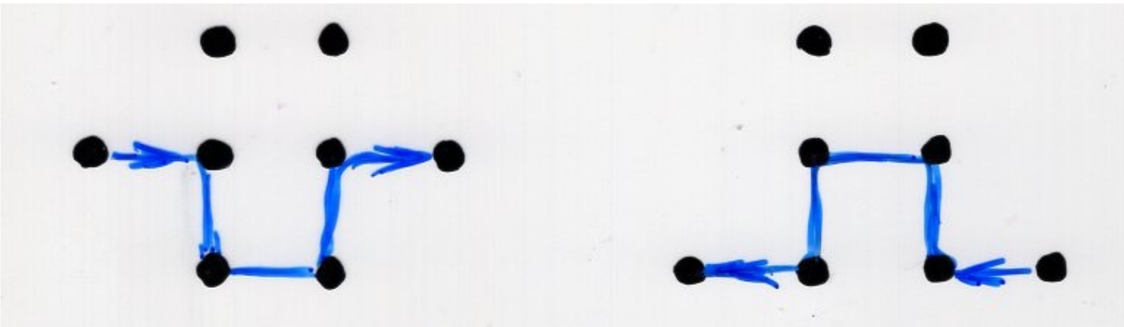
spanning tree

two amazing facts



same probability law on $\overrightarrow{\eta}$ and $\overleftarrow{\eta}$





graph $G = (V, E)$ $\quad V = \{s_1, s_2, \ldots\}$

w valuation $\quad A = (a_{ij})$

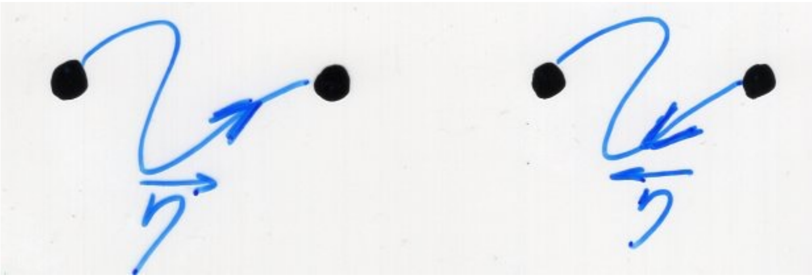
$w(s_i, s_j) = a_{ij}$

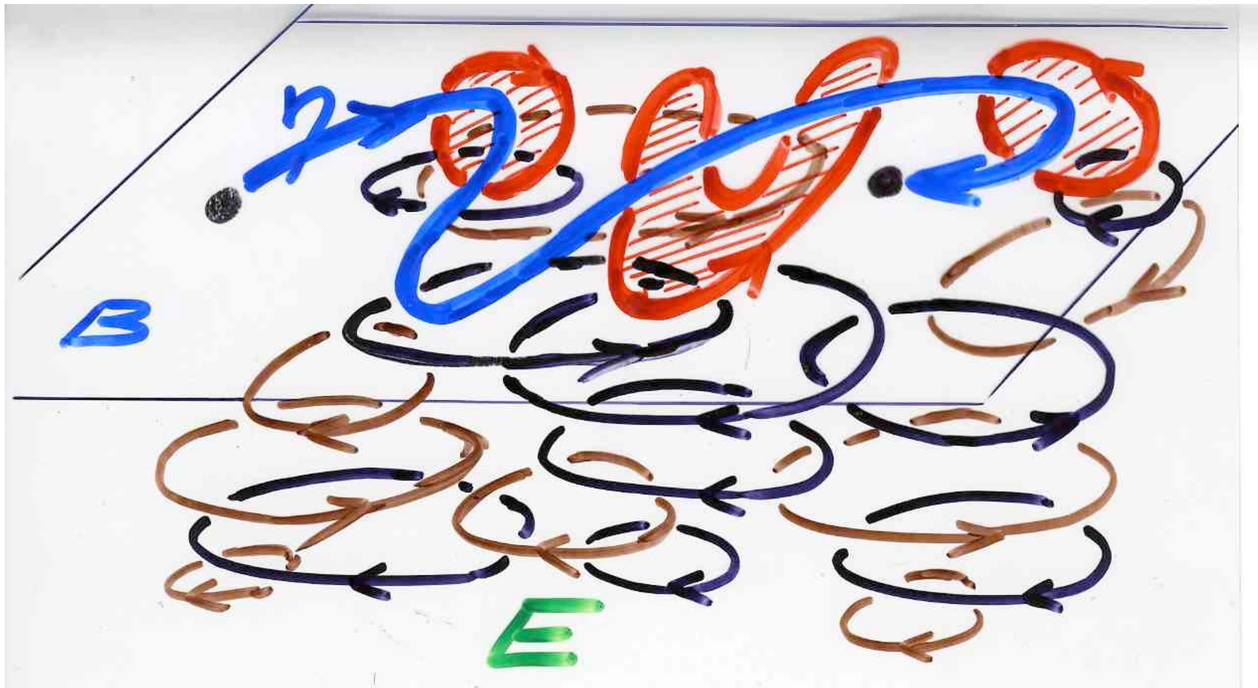
ω walk (path) on G $\quad w(\omega)$

ω
$u \rightsquigarrow v$
$u, v \in V$ $\quad \xrightarrow[\text{loops}]{\text{erasing}}$ self-avoiding walk η
$u \rightsquigarrow v$

$$V(\eta) = \sum_{\substack{\omega \\ u \rightsquigarrow v \\ \omega \to \eta}} w(\omega)$$

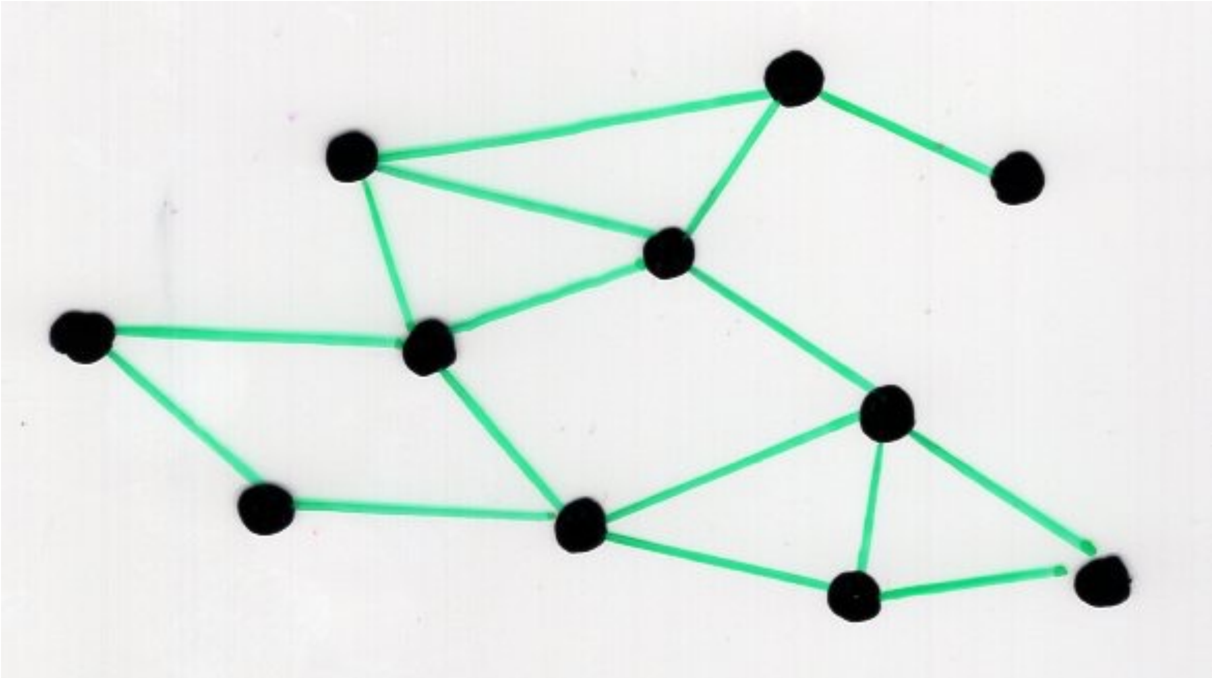
The advantage of ... "organic" combinatorics



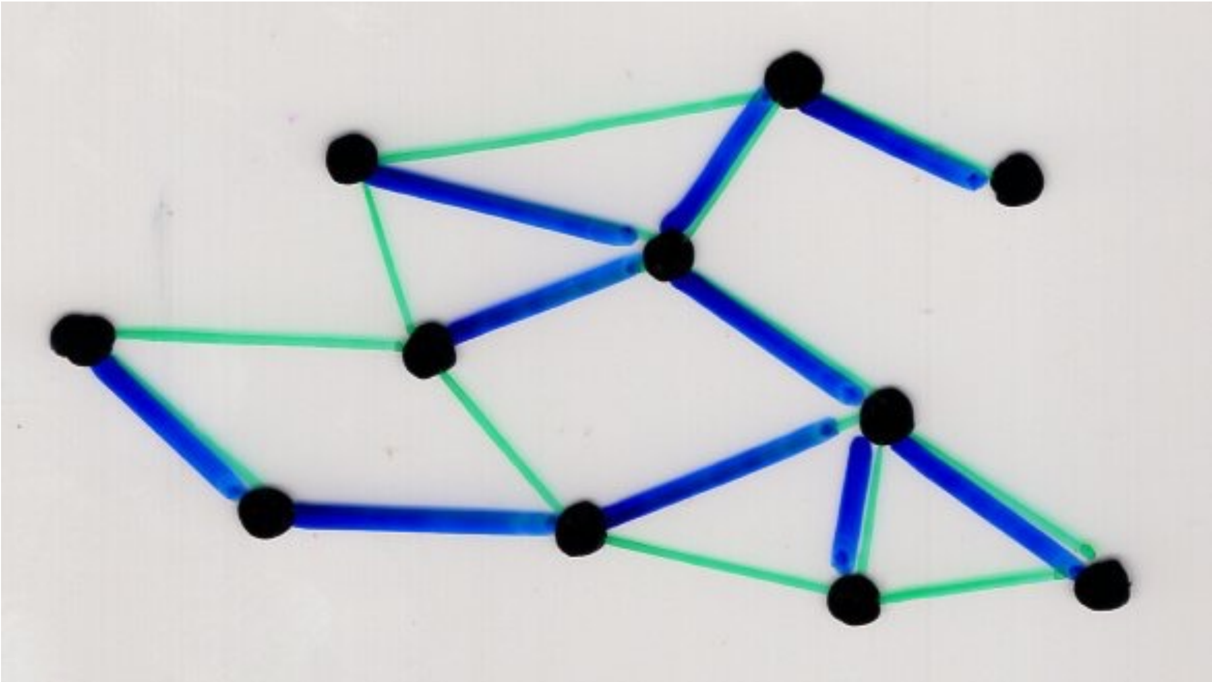


same probability law on $\vec{\eta}$ and $\overleftarrow{\eta}$

spanning tree
of a graph $G = (V, E)$

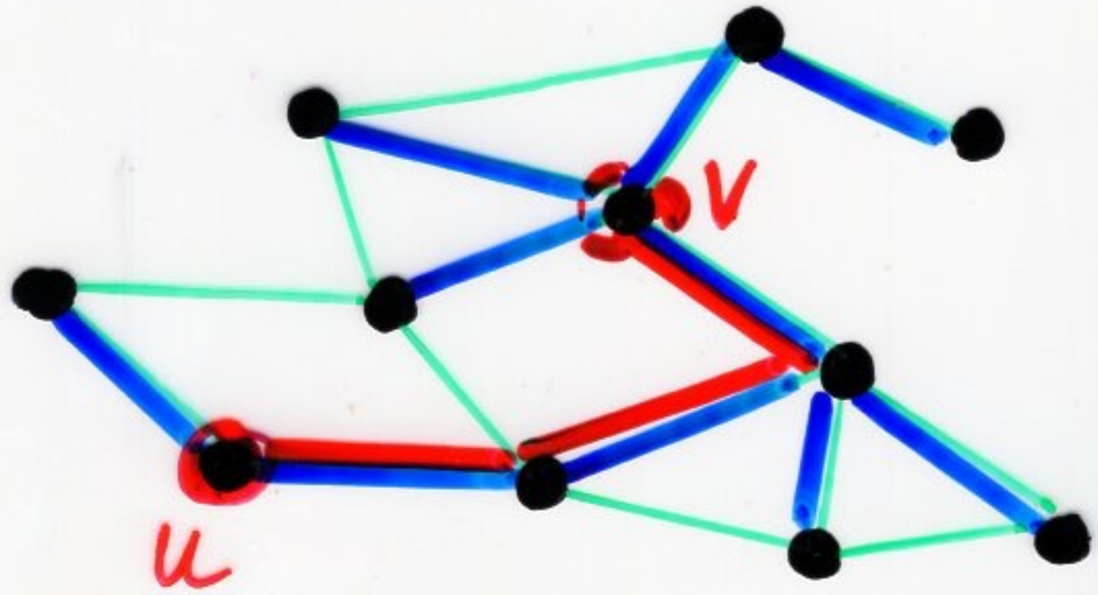


spanning tree
of a graph $G = (V, E)$



spanning tree T of G
with uniform probability

spanning tree
of a graph $G = (V, E)$



$u, v \in V$
unique path ω
$u \rightsquigarrow v$ on the tree T

spanning tree T of G
with uniform probability

same
probability law
as a LERW
$u \rightsquigarrow v$ on G

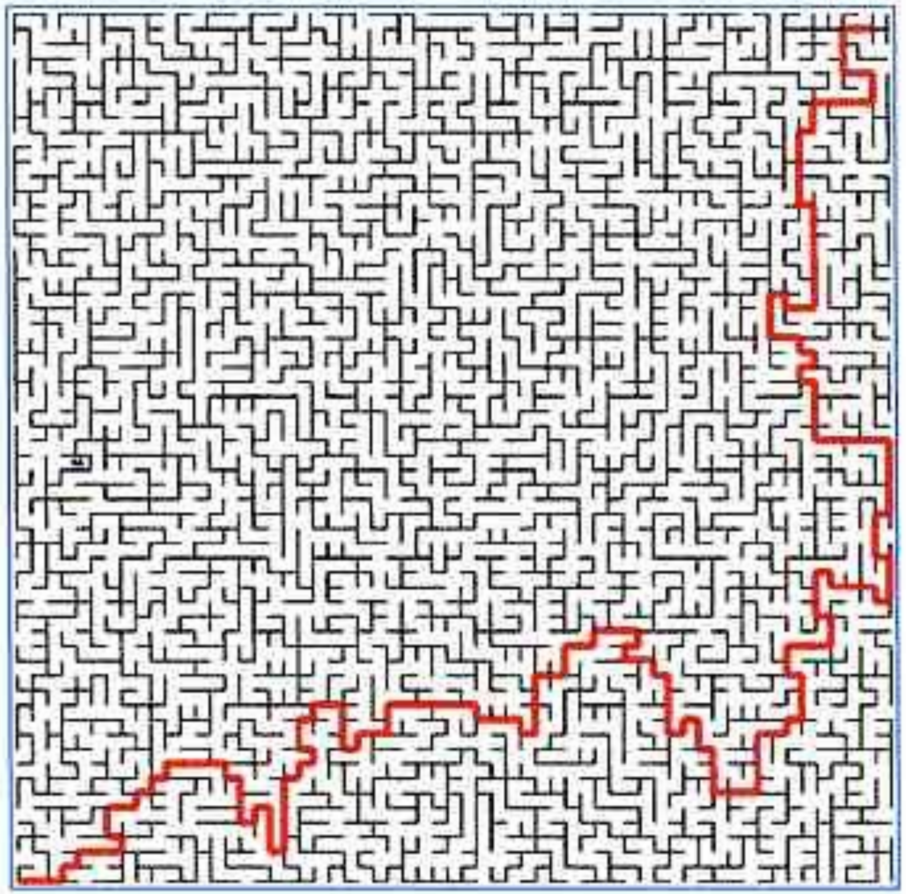
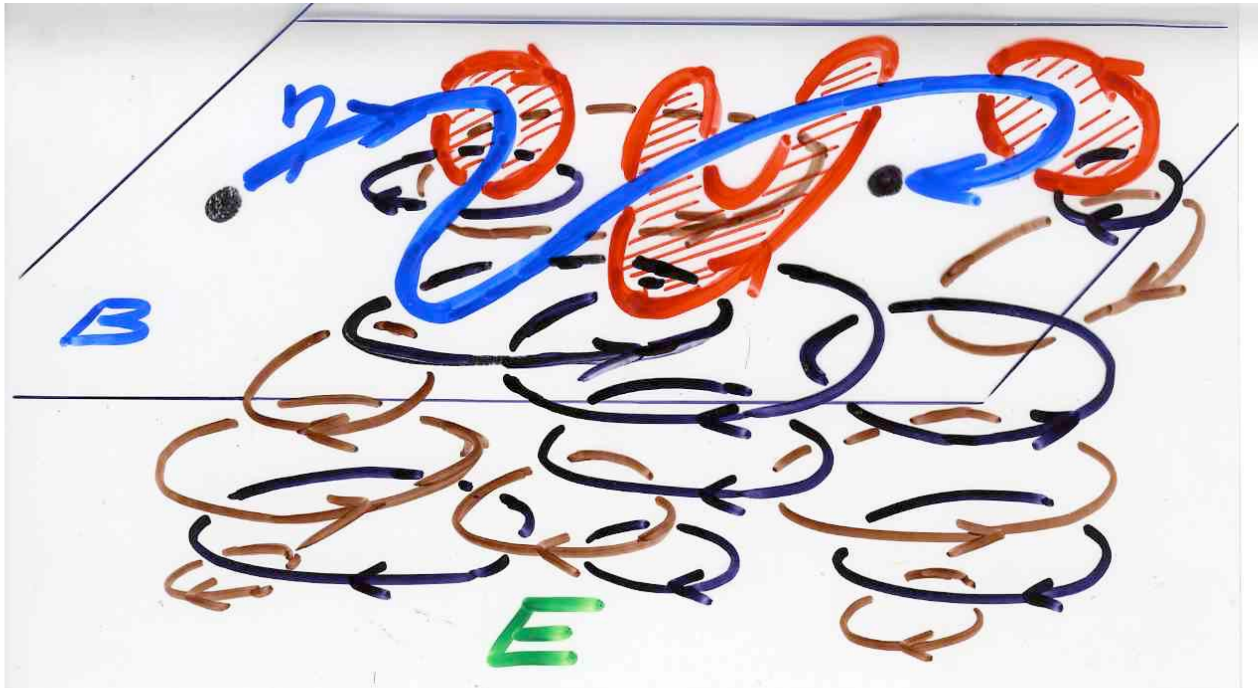


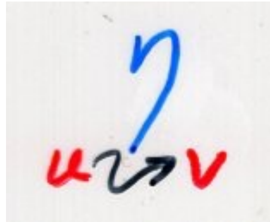
Figure 1.1: The LERW in the UST.

for path $\omega \longrightarrow f(\omega) \in F(X)$

- what do you "see" above $f(\omega)$
- " " " below $f(\omega)$

$$\omega \overset{\chi}{\longleftrightarrow} (\eta, E)$$

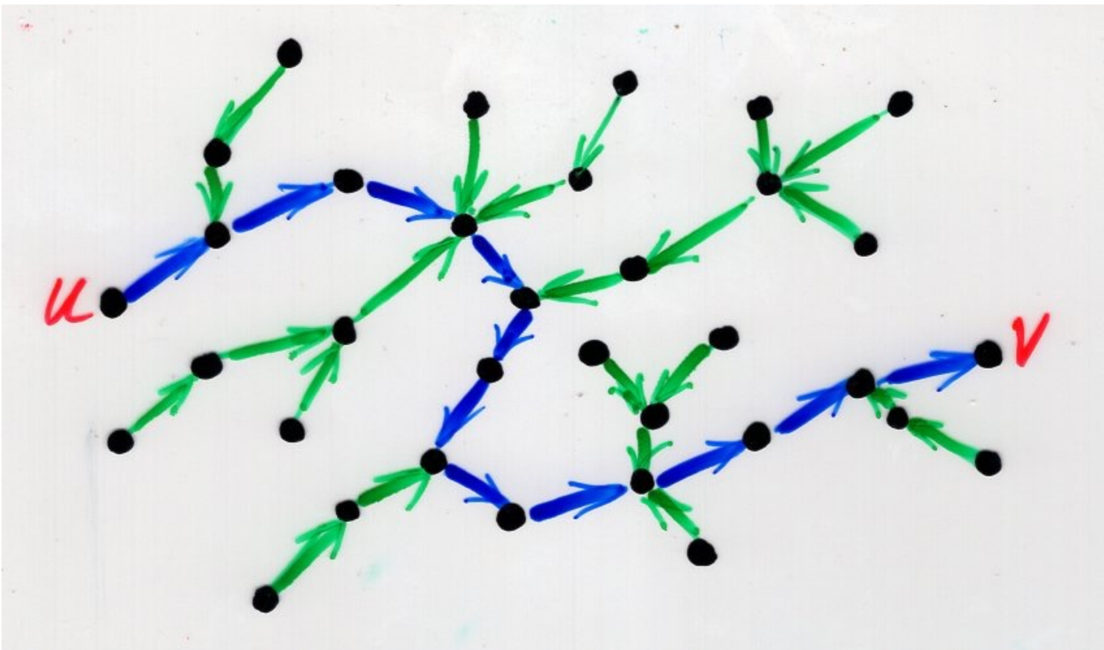




for path $\omega \longrightarrow f(\omega) \in F(X)$

- what do you "see" above $f(\omega)$
- " " " below $f(\omega)$

$\omega \not\longleftrightarrow (\eta, E) \longrightarrow T(\omega)$ spanning tree on $\mathrm{supp}(\omega)$



$$\omega \overset{\chi}{\longleftrightarrow} (\eta, E) \longrightarrow T(\omega)$$

spanning tree on $\mathrm{supp}(\omega)$

ω $u \rightsquigarrow v$

η $u \rightsquigarrow v$

research problem 1

- Prove the equivalence

$$\begin{cases} \cdot \text{ UST uniform spanning tree} \\ \cdot \text{ LERW for } \eta_{u \rightsquigarrow v} \end{cases}$$

using the theory of heaps

- Is $T(\omega)$ a UST on $\mathrm{supp}(\omega)$?

complements

Wilson's algorithm
for
uniform random spanning tree

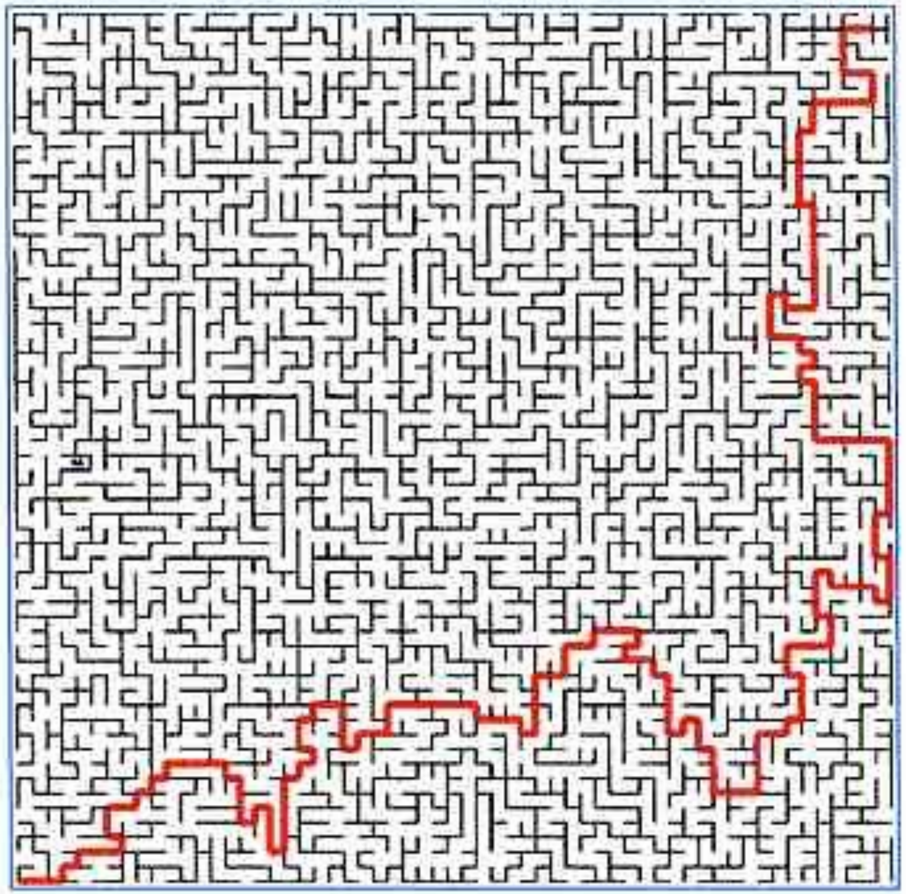
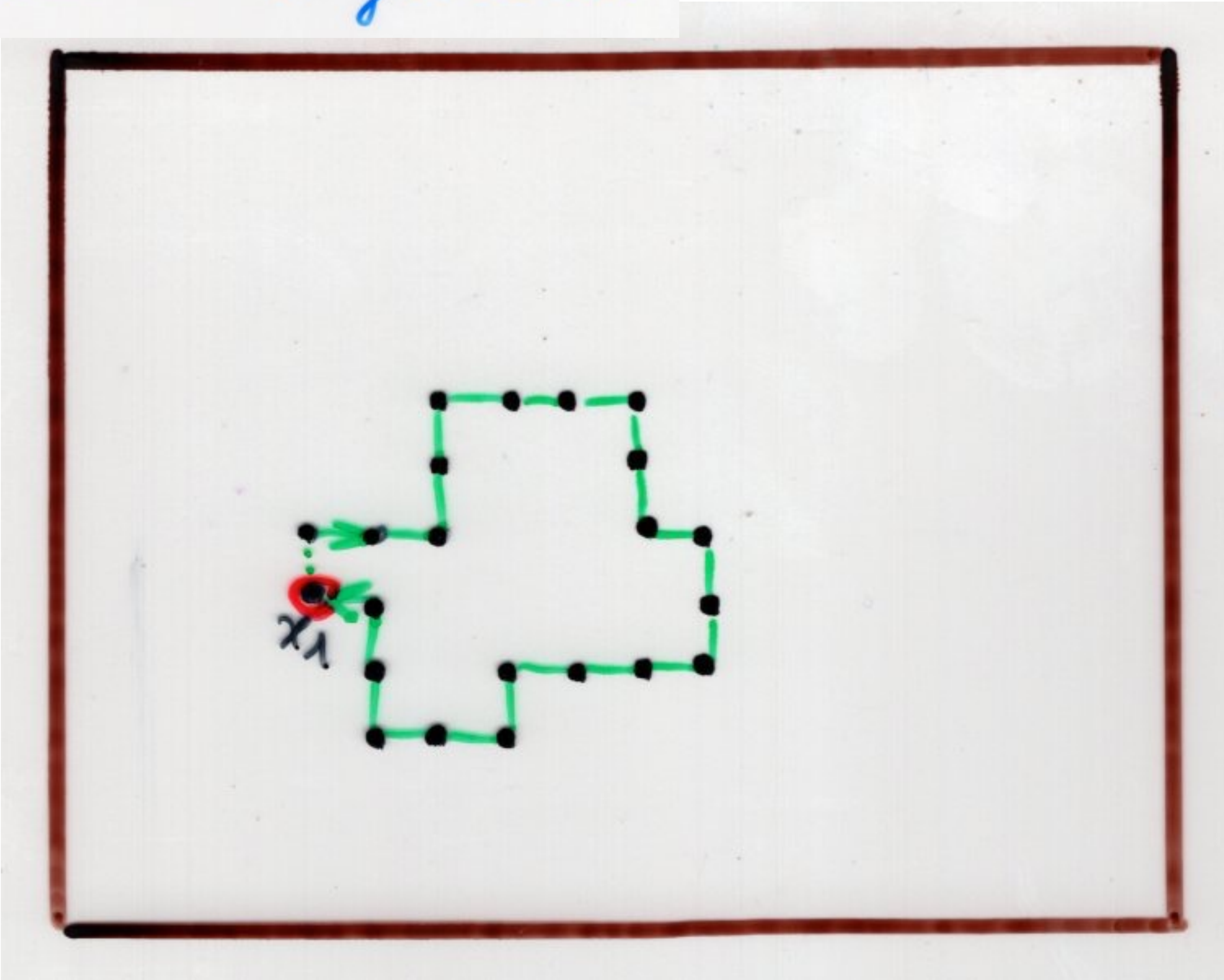
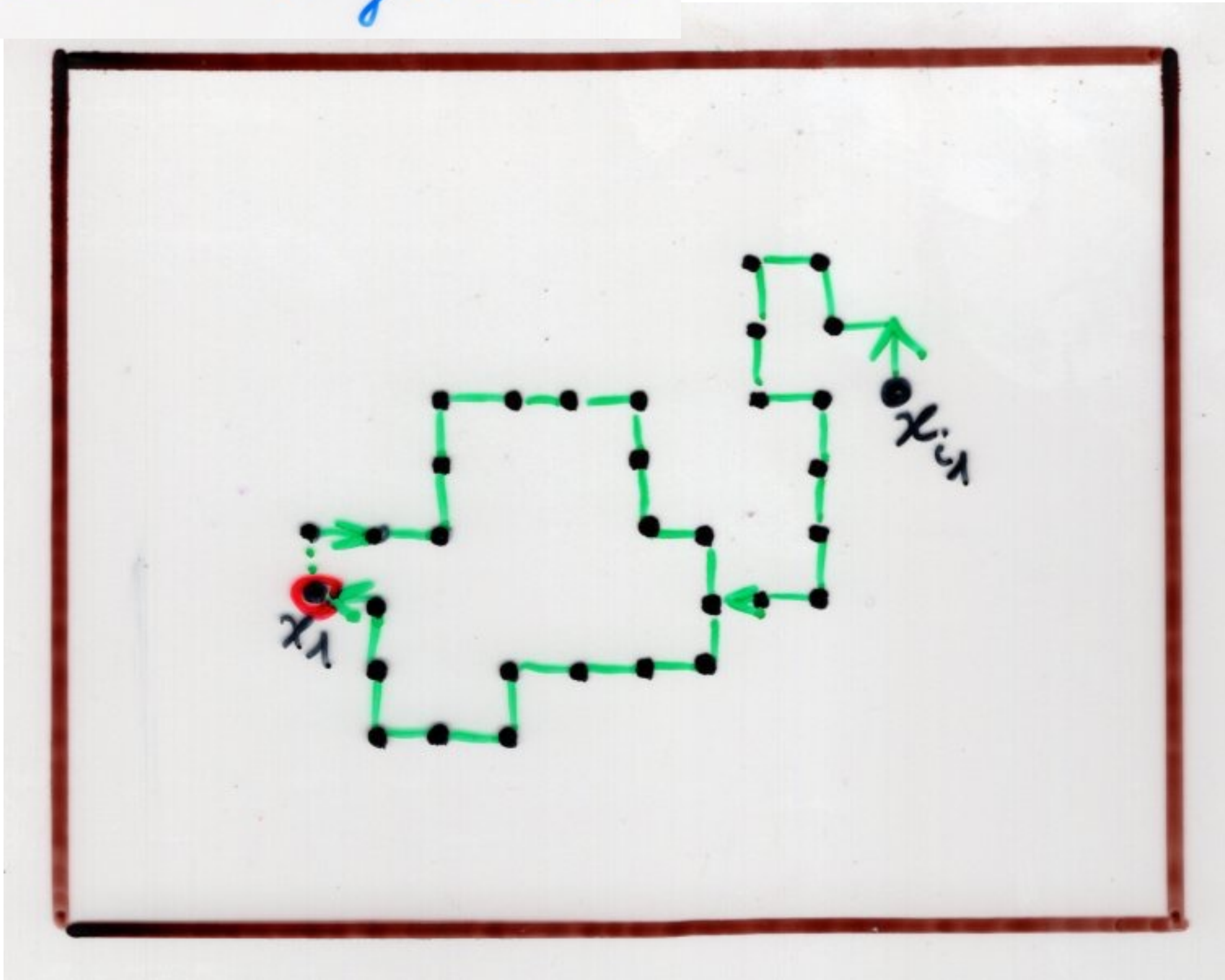


Figure 1.1: The LERW in the UST.

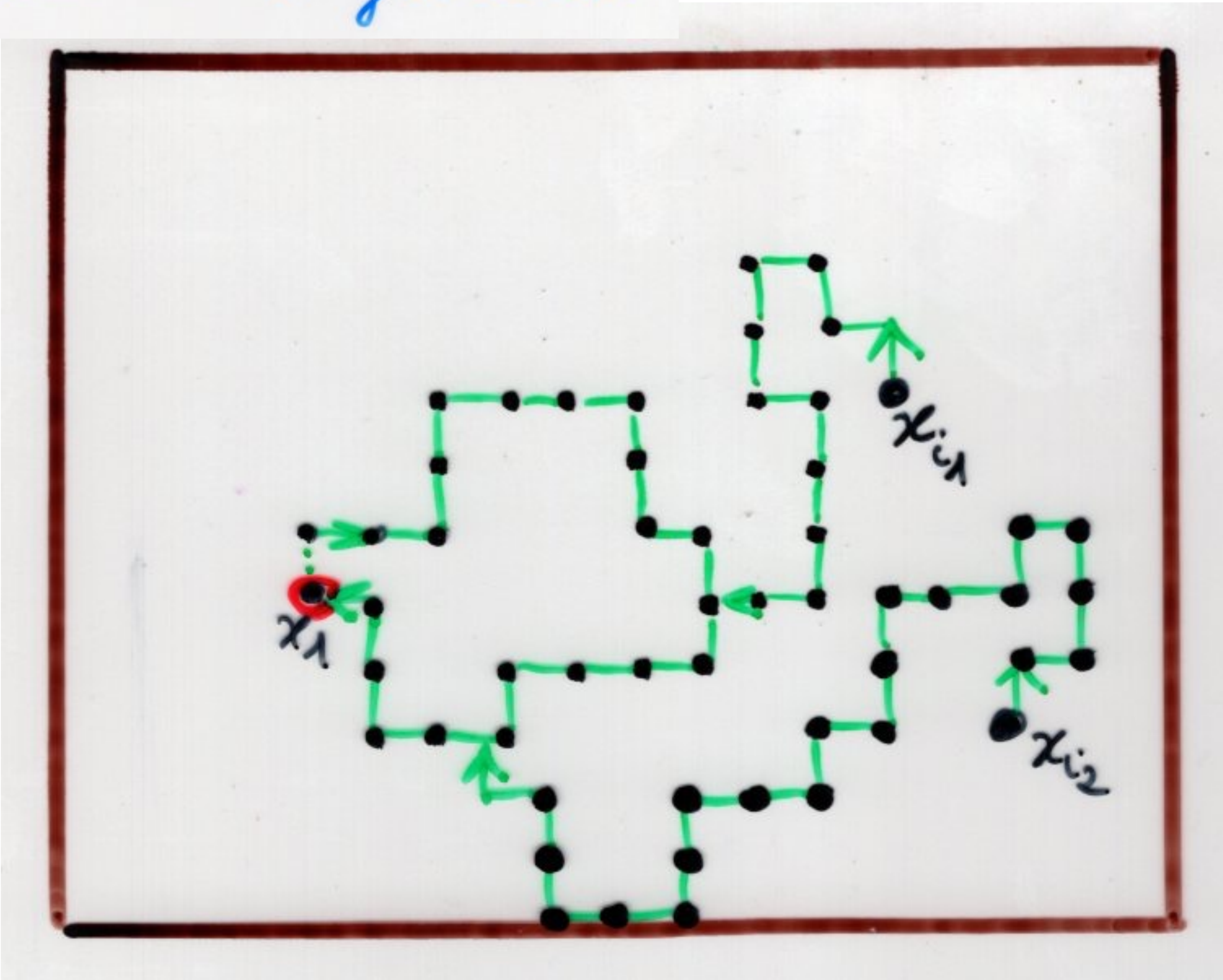
Wilson's algorithm



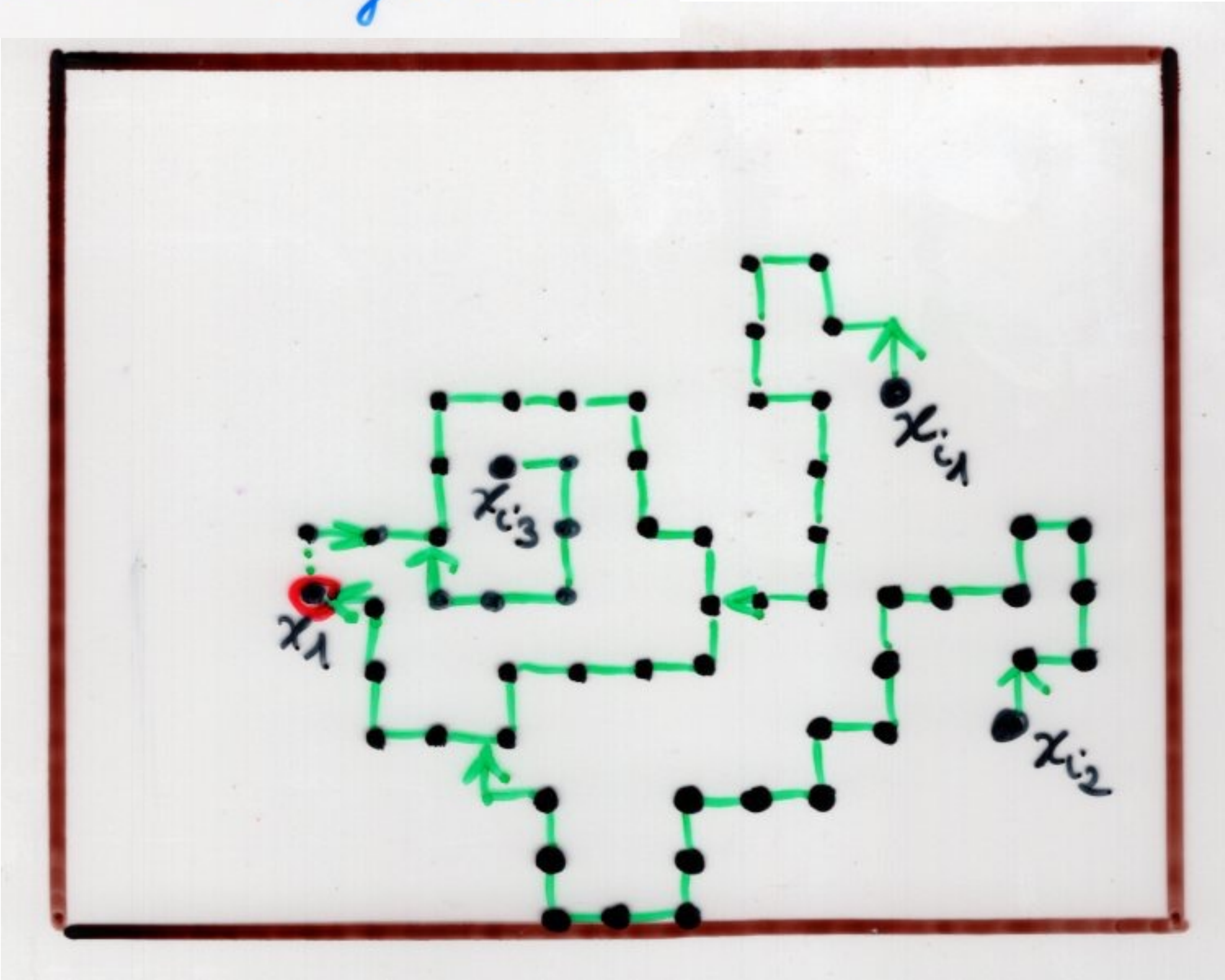
Wilson's algorithm



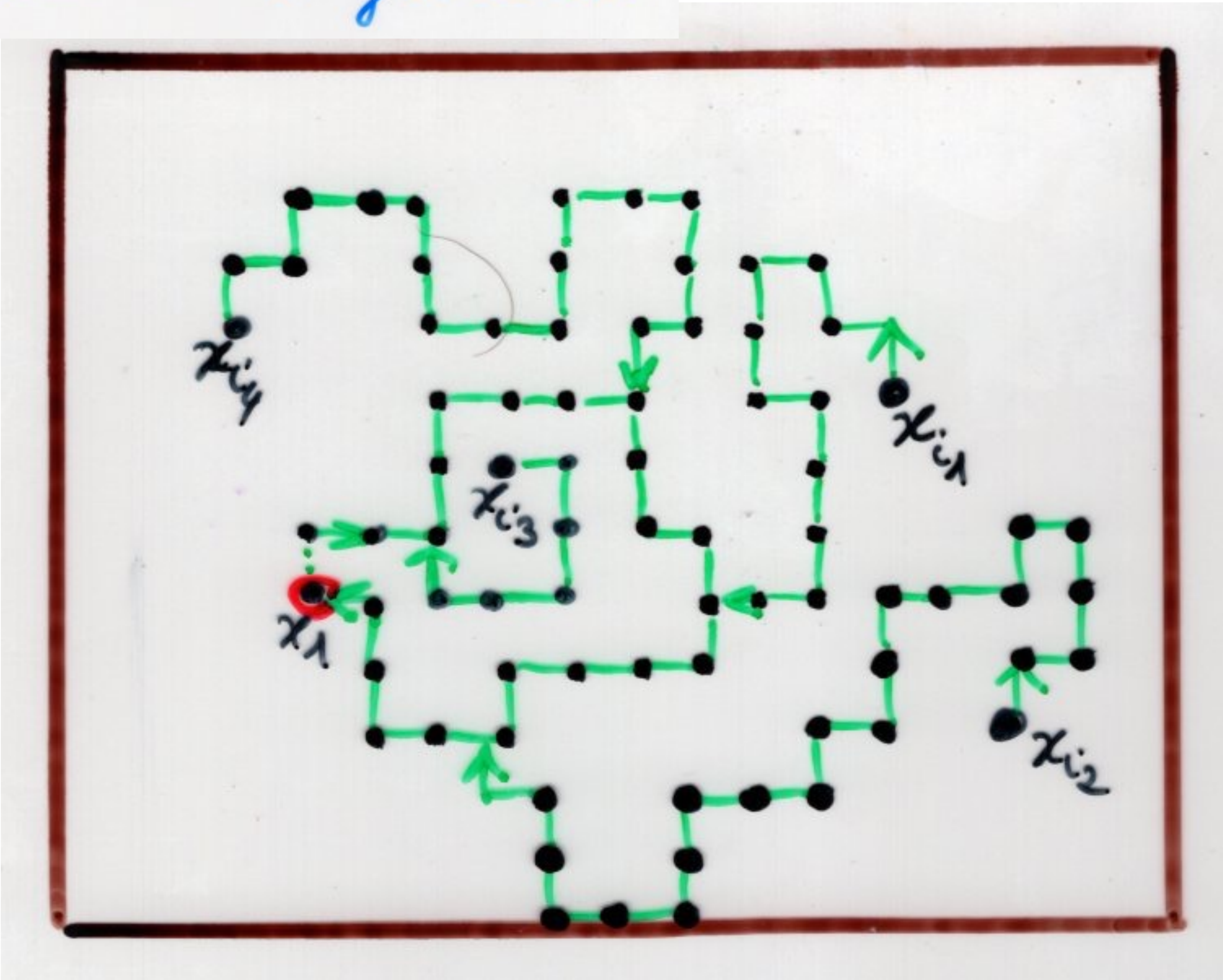
Wilson's algorithm



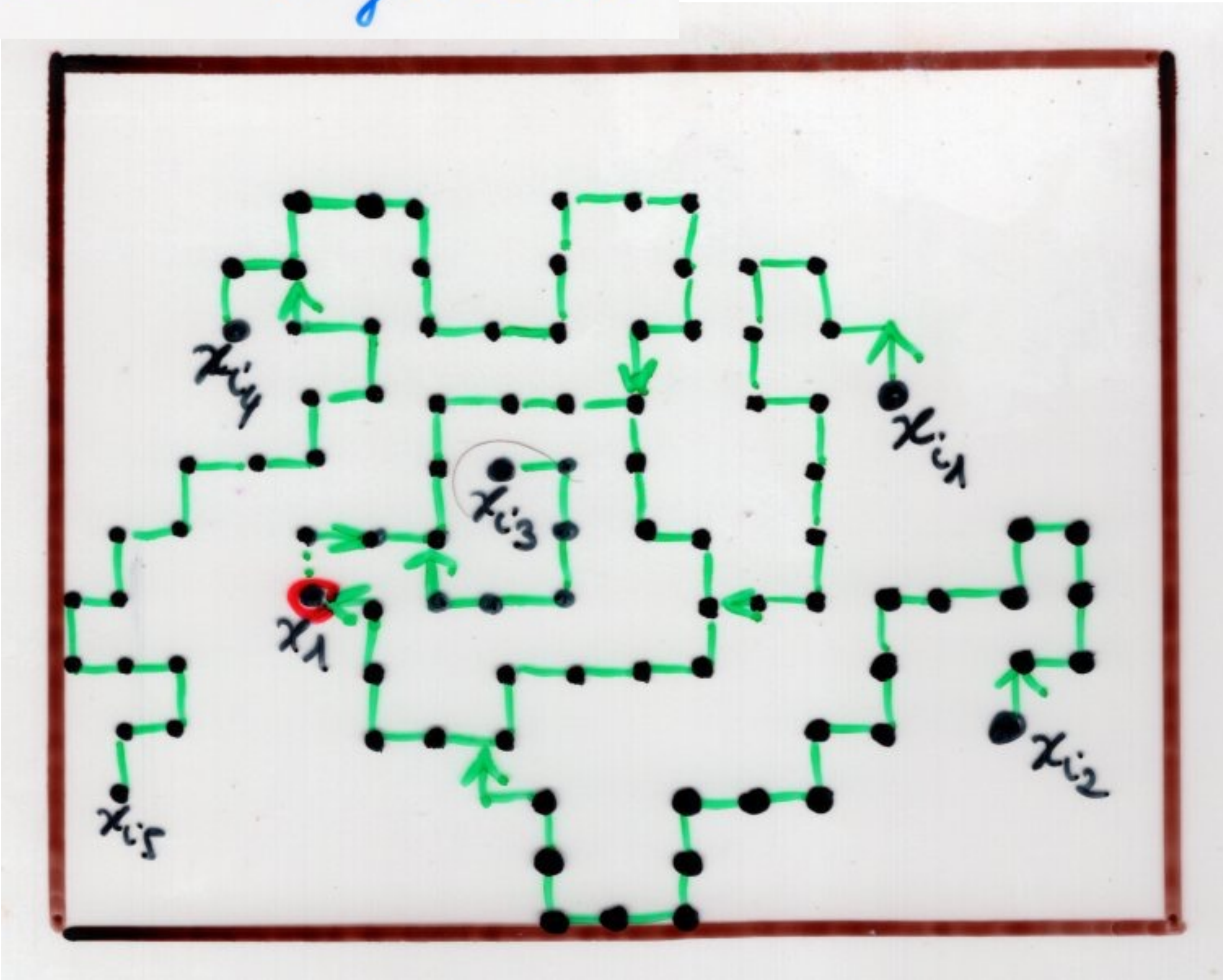
Wilson's algorithm



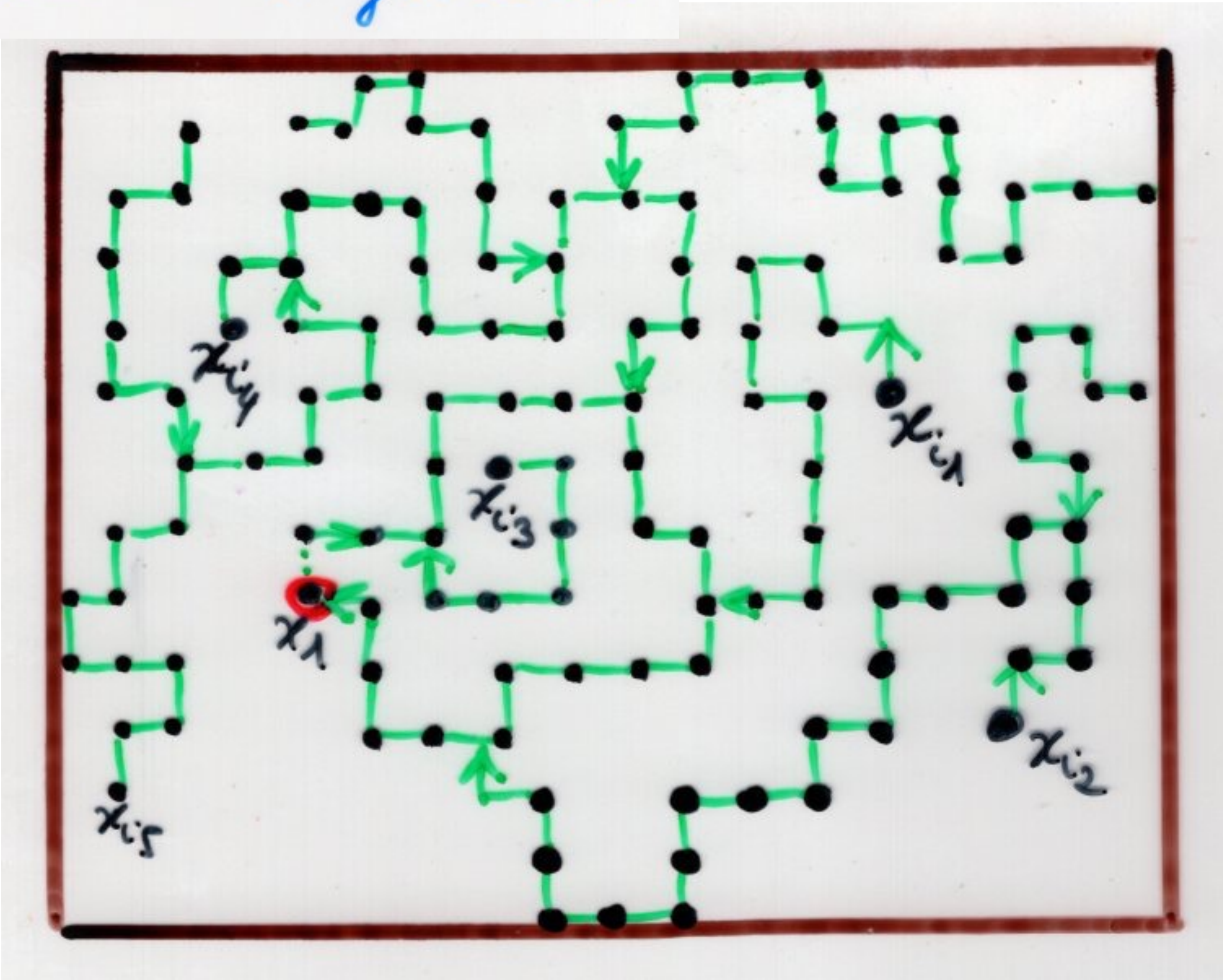
Wilson's algorithm



Wilson's algorithm



Wilson's algorithm



Wilson's algorithm

animation: see the video

by Mike Rostock

https://bl.ocks.org/mbostock/11357811

Research problem 2

- Take a random path ω_T on $G = (X, E)$ graph starting at $u \in X$.

$$\omega_T = (s_0 = u, \dots, s_T)$$

$$\omega_T \longrightarrow (\eta_T ; E_T)$$

- for each $T = 0, 1, ..$ we get a spanning tree $\mathbf{T}_T$ of the support of ω_T by taking the maximal edges of the flow F_T

$$F_T = f(\omega)_T = f(\eta_T) f(E_T)$$

- when the support of ω_T is X, we stop and get a spanning tree of X

- when the support of ω_T is X, we stop and get a spanning tree of X

a) Is it a uniform random spanning tree of $G = (X, E)$?

b) Compare this "organic" algorithm with Wilson's algorithm