# Random generation of combinatorial structures

## Uniform random maps and graphs on surfaces using Boltzmann sampling

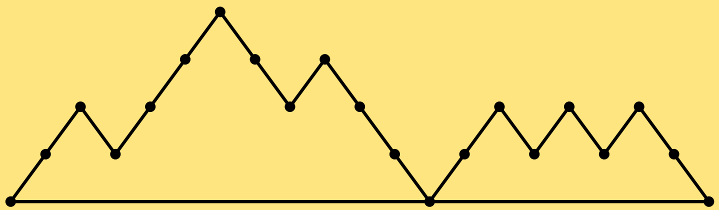
a survey by Gilles Schaeffer

CNRS / Ecole Polytechnique,
Palaiseau, France

# What is uniform random generation?

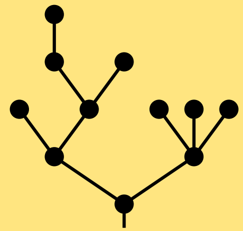
A combinatorial class $\mathcal{A}$, ranked by a size: $\mathcal{A}_n = \{a \in \mathcal{A}, |a| = n\}$ finite.

Ex: balanced parenthesis words ($n$ pairs) or ordered trees ($n$ edges)



$(()((())()))(()()())$

$\Leftrightarrow$

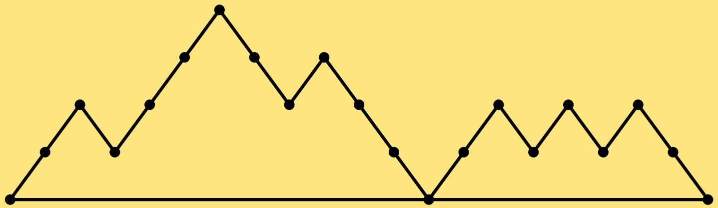


$$|\mathcal{A}_n| = \frac{1}{n+1}\binom{2n}{n}$$

# What is uniform random generation?

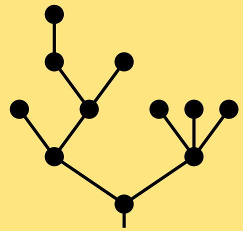
A combinatorial class $\mathcal{A}$, ranked by a size: $\mathcal{A}_n = \{a \in \mathcal{A}, |a| = n\}$ finite.

Ex: balanced parenthesis words ($n$ pairs) or ordered trees ($n$ edges)



$(()((())()))(()()())$

$\Leftrightarrow$



$$|\mathcal{A}_n| = \frac{1}{n+1}\binom{2n}{n}$$

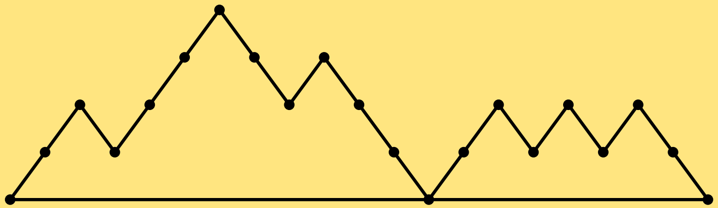
Uniform random sampler $\mathsf{U}\mathcal{A}(n)$ on $\mathcal{A}_n$:

$$\mathrm{Pr}(\mathsf{U}\mathcal{A}(n) = t) = \frac{1}{|\mathcal{A}_n|}, \text{ for any } t \in \mathcal{A}_n.$$

# What is uniform random generation?

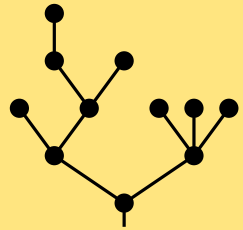
A combinatorial class $\mathcal{A}$, ranked by a size: $\mathcal{A}_n = \{a \in \mathcal{A}, |a| = n\}$ finite.

Ex: balanced parenthesis words ($n$ pairs) or ordered trees ($n$ edges)



$(()((())()))(()()())$

$\Leftrightarrow$



$$|\mathcal{A}_n| = \frac{1}{n+1}\binom{2n}{n}$$

Uniform random sampler $\mathsf{U}\mathcal{A}(n)$ on $\mathcal{A}_n$:

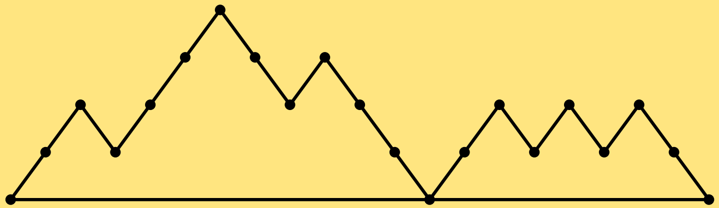
$$\Pr(\mathsf{U}\mathcal{A}(n) = t) = \frac{1}{|\mathcal{A}_n|}, \text{ for any } t \in \mathcal{A}_n.$$

Ex: via Catalan's bijection: apply a random permutation to $({}^n)^n$ to get a uniform random word among the $\binom{2n}{n}$ parenthesis words

# What is uniform random generation?

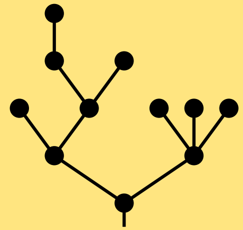
A combinatorial class $\mathcal{A}$, ranked by a size: $\mathcal{A}_n = \{a \in \mathcal{A}, |a| = n\}$ finite.

Ex: balanced parenthesis words ($n$ pairs) or ordered trees ($n$ edges)



$(()((())()))(()()())$

$\Leftrightarrow$

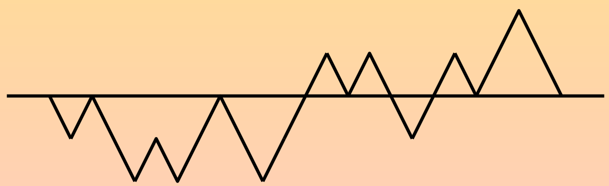


$$|\mathcal{A}_n| = \frac{1}{n+1}\binom{2n}{n}$$

Uniform random sampler $\mathsf{U}\mathcal{A}(n)$ on $\mathcal{A}_n$:

$$\Pr(\mathsf{U}\mathcal{A}(n) = t) = \frac{1}{|\mathcal{A}_n|}, \text{ for any } t \in \mathcal{A}_n.$$

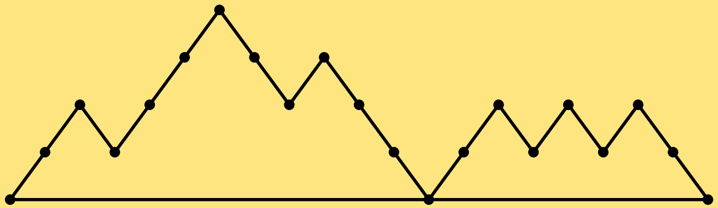
Ex: via Catalan's bijection: apply a random permutation to $(^n)^n$ to get a uniform random word among the $\binom{2n}{n}$ parenthesis words



# What is uniform random generation?

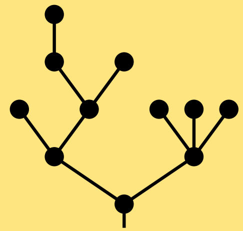
A combinatorial class $\mathcal{A}$, ranked by a size: $\mathcal{A}_n = \{a \in \mathcal{A}, |a| = n\}$ finite.

Ex: balanced parenthesis words ($n$ pairs) or ordered trees ($n$ edges)



$(()((())()))(()()())$

$\Leftrightarrow$

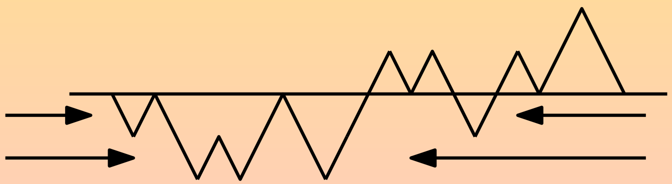


$$|\mathcal{A}_n| = \tfrac{1}{n+1}\tbinom{2n}{n}$$

Uniform random sampler $\mathsf{U}\mathcal{A}(n)$ on $\mathcal{A}_n$:

$$\Pr(\mathsf{U}\mathcal{A}(n) = t) = \tfrac{1}{|\mathcal{A}_n|}, \text{ for any } t \in \mathcal{A}_n.$$

Ex: via Catalan's bijection: apply a random permutation to $({}^n)^n$ to get a uniform random word among the $\binom{2n}{n}$ parenthesis words

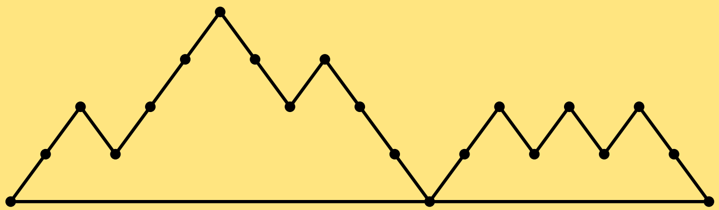


$\Rightarrow$

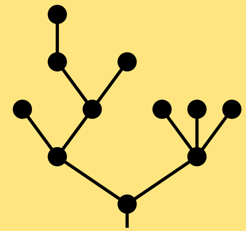
# What is uniform random generation?

A combinatorial class $\mathcal{A}$, ranked by a size: $\mathcal{A}_n = \{a \in \mathcal{A}, |a| = n\}$ finite.

Ex: balanced parenthesis words ($n$ pairs) or ordered trees ($n$ edges)



$\Leftrightarrow$



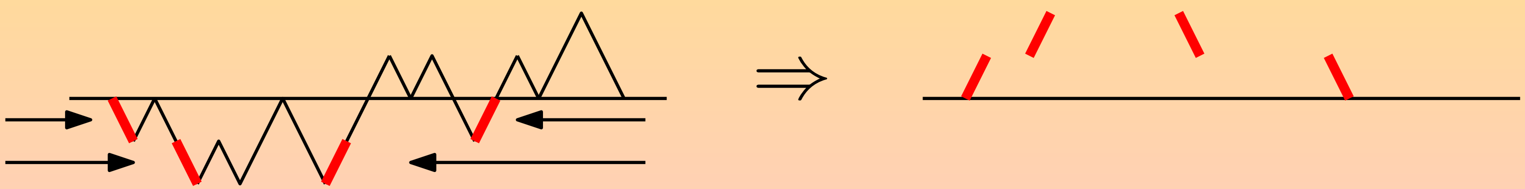
$$|\mathcal{A}_n| = \frac{1}{n+1}\binom{2n}{n}$$

$$(()((())()))(()()())$$

Uniform random sampler $\mathsf{U}\mathcal{A}(n)$ on $\mathcal{A}_n$:

$$\Pr(\mathsf{U}\mathcal{A}(n) = t) = \frac{1}{|\mathcal{A}_n|}, \text{ for any } t \in \mathcal{A}_n.$$

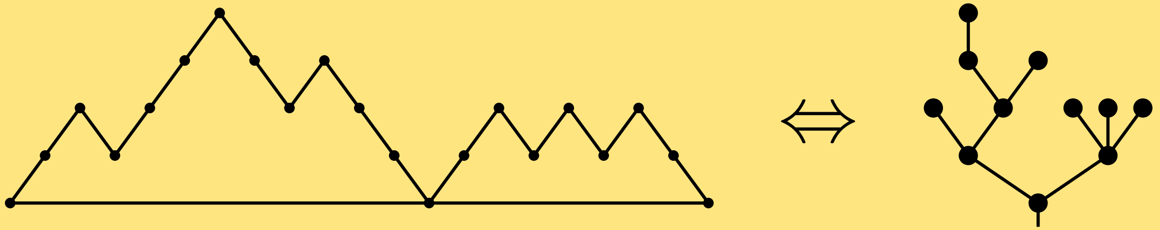
Ex: via Catalan's bijection: apply a random permutation to $(^n)^n$ to get a uniform random word among the $\binom{2n}{n}$ parenthesis words



# What is uniform random generation?

A combinatorial class $\mathcal{A}$, ranked by a size: $\mathcal{A}_n = \{a \in \mathcal{A}, |a| = n\}$ finite.

Ex: balanced parenthesis words ($n$ pairs) or ordered trees ($n$ edges)



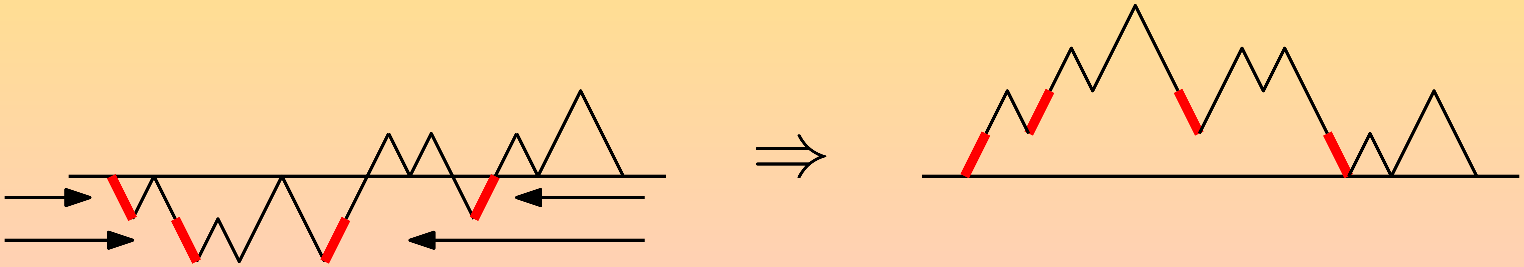
$(()((())()))(()()())$

$$|\mathcal{A}_n| = \tfrac{1}{n+1}\tbinom{2n}{n}$$

Uniform random sampler $\mathsf{U}\mathcal{A}(n)$ on $\mathcal{A}_n$:

$$\Pr(\mathsf{U}\mathcal{A}(n) = t) = \tfrac{1}{|\mathcal{A}_n|}, \text{ for any } t \in \mathcal{A}_n.$$

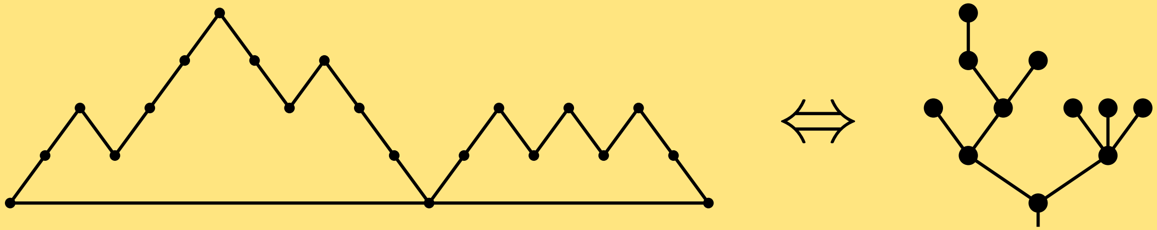
Ex: via Catalan's bijection: apply a random permutation to $({}^n)^n$ to get a uniform random word among the $\binom{2n}{n}$ parenthesis words



# What is uniform random generation?

A combinatorial class $\mathcal{A}$, ranked by a size: $\mathcal{A}_n = \{a \in \mathcal{A}, |a| = n\}$ finite.

Ex: balanced parenthesis words ($n$ pairs) or ordered trees ($n$ edges)



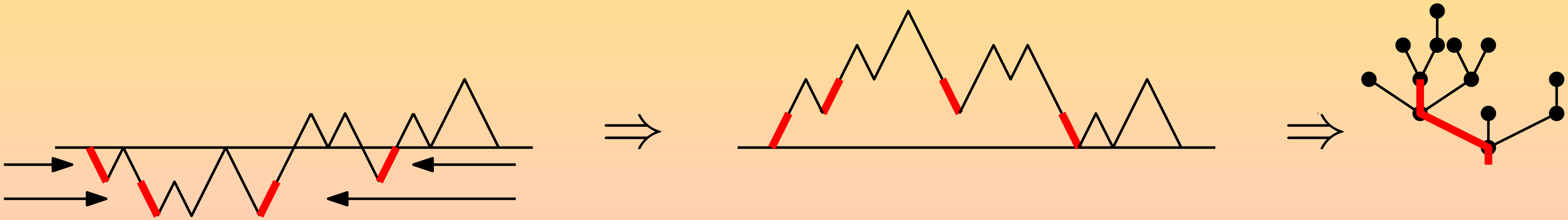
$(()((())()))(()()())$

$$|\mathcal{A}_n| = \frac{1}{n+1}\binom{2n}{n}$$

Uniform random sampler $\mathsf{U}\mathcal{A}(n)$ on $\mathcal{A}_n$:

$$\Pr(\mathsf{U}\mathcal{A}(n) = t) = \frac{1}{|\mathcal{A}_n|}, \text{ for any } t \in \mathcal{A}_n.$$

Ex: via Catalan's bijection: apply a random permutation to $({}^n)^n$ to get a uniform random word among the $\binom{2n}{n}$ parenthesis words



## Uniform random generation, what for?

In silico combinatorics, statistical physics and biology

too many people involved... Denise, Ponty

Experimental companion to average case analysis of algorithms

Flajolet, Zimmermann

Statistical test in model checking

Gaudel, Gouraud, Denise

Average drawing size analysis for planar drawing algorithms

Fusy, S.

## Random sampling paradigms

Markov chain simulations: venerable topic $\longrightarrow$ perfect sampling
$\Rightarrow$ versatile but slow in general (polynomial is good)

Recursive sampling: requires a "combinatorial" recurrence
$\Rightarrow$ optimal when nice bijections are available (linear is good)
$\Rightarrow$ ok for all "decomposable" structures (quadratic is good)

Boltzmann sampling: replace exact counting by GF evaluation
$\Rightarrow$ efficient for decomposable structures and more (linear/quad)

We concentrate on Boltzmann sampling...

## Boltzmann models, Boltzmann sampling

A combinatorial class $\mathcal{A} = (\mathcal{A}_n)_{n\geq 0}$

Its generating function $A(x) = \sum_{a\in A} x^{|a|} = \sum_n |\mathcal{A}_n| x^n$.

Let $x_0 > 0$ be such that $A(x_0)$ is finite (e.g. $x_0 < \rho_A$)

$\Gamma[\mathcal{A}](x_0)$ is a Boltzmann generator of parameter $x_0$ for $\mathcal{A}$ if

$$\Pr(\Gamma[\mathcal{A}](x_0) = a) = \frac{x^{|a|}}{A(x)} \text{ for all } x \in \mathcal{A}.$$

Boltzmann generators are compatible with the sum, product and composition of combinatorial classes.

$$\Gamma[\mathcal{A}+\mathcal{B}](x) := \texttt{if}\ \mathrm{Bern}\left(\frac{A(x)}{A(x)+B(x)}\right) \texttt{then}\, \Gamma[\mathcal{A}](x)\, \texttt{else}\, \Gamma[\mathcal{B}](x)$$

$$\Gamma[\mathcal{A}\times\mathcal{B}](x) := (\Gamma[\mathcal{A}(x)], \Gamma[\mathcal{B}(x)])$$

$$\Gamma[\mathcal{A}\circ\mathcal{B}](x) := \texttt{let}\, a = \Gamma[\mathcal{A}](B(x))\, \texttt{in}\, (a; (\Gamma[\mathcal{B}](x))^{|a|})$$

## Composition in Boltzmann sampling

$\Gamma[\mathcal{A} \circ \mathcal{B}](x) := \texttt{let}\, a = \Gamma[\mathcal{A}](B(x))\, \texttt{in}\, (a; (\Gamma[\mathcal{B}](x))^{|a|})$

Theorem: if $\Gamma[\mathcal{A}]$ and $\Gamma[\mathcal{B}]$ are Boltzmann so is $\Gamma[\mathcal{A} \circ \mathcal{B}]$.

Proof: Let $\gamma \in A \circ B$ with $\gamma = (a; b_1, \ldots, b_k)$ where $a \in \mathcal{A}$, $k = |a|$, $b_i \in \mathcal{B}$ for $i = 1, \ldots, k$, and $|\gamma| = |b_1| + \ldots + |b_k|$.

Then $\Pr\left(\Gamma[\mathcal{A} \circ \mathcal{B}](x) = \gamma\right)$

$$= \Pr\left(\Gamma[\mathcal{A}] = a\right) \cdot \prod_{i=1}^{|a|} \Pr\left(\Gamma[\mathcal{B}](x) = b_i\right)$$

$$= \frac{B(x)^{|a|}}{A(B(x))} \cdot \frac{\prod_i x^{|b_i|}}{B(x)^{|a|}} = \frac{x^{|b_1|+\cdots+|b_k|}}{A(B(x))} = \frac{x^{|\gamma|}}{(A \circ B)(x)}. \qquad \square$$

Theorem: if $\Gamma[\mathcal{A} \circ \mathcal{B}]$ is Boltzmann then so are Core$(\Gamma[\mathcal{A} \circ \mathcal{B}])$ and First$(\Gamma[\mathcal{A} \circ \mathcal{B}])$, where Core$(\gamma) = a$ and First$(\gamma) = b_1$.

# An example: Boltzmann for trees

Recall that $\mathcal{A}$ is the familly of ordered trees: a tree decomposes into a root and a sequence of subtrees attached by edges:

$$\mathcal{A} = \{\textsf{root}\} \times \mathrm{Seq}(\{\textsf{edge}\} \times \mathcal{A})$$

$$\Gamma[\mathcal{A}](x) := \ \texttt{let}\, k = |\Gamma[\textsf{Seq}](xA(x))| \ \texttt{in} \ \left(\textsf{root}; (\{\textsf{edge}\} \times \Gamma[\mathcal{A}](x))^k\right)$$

where the size of a random sequence under the Boltzmann model simply follows a geometric law: $\Pr(|\Gamma[\textsf{Seq}](p)| = k) = p^k(1-p)$.

# An example: Boltzmann for trees

Recall that $\mathcal{A}$ is the familly of ordered trees: a tree decomposes into a root and a sequence of subtrees attached by edges:

$$\mathcal{A} = \{\textsf{root}\} \times \mathrm{Seq}(\{\textsf{edge}\} \times \mathcal{A})$$

$$\Gamma[\mathcal{A}](x) := \texttt{ let}\, k = |\Gamma[\textsf{Seq}](xA(x))| \texttt{ in } \left(\textsf{root}; (\{\textsf{edge}\} \times \Gamma[\mathcal{A}](x))^k\right)$$

where the size of a random sequence under the Boltzmann model simply follows a geometric law: $\Pr(|\Gamma[\textsf{Seq}](p)| = k) = p^k(1-p)$.

# An example: Boltzmann for trees

Recall that $\mathcal{A}$ is the familly of ordered trees: a tree decomposes into a root and a sequence of subtrees attached by edges:

$$\mathcal{A} = \{\mathsf{root}\} \times \mathrm{Seq}(\{\mathsf{edge}\} \times \mathcal{A})$$

$$\Gamma[\mathcal{A}](x) := \ \mathtt{let}\, k = |\Gamma[\mathsf{Seq}](xA(x))|\ \mathtt{in}\ \big(\mathsf{root}; (\{\mathsf{edge}\} \times \Gamma[\mathcal{A}](x))^k\big)$$

where the size of a random sequence under the Boltzmann model simply follows a geometric law: $\Pr(|\Gamma[\mathsf{Seq}](p)| = k) = p^k(1-p)$.

$\Gamma[\mathsf{Seq}] = 3$ 

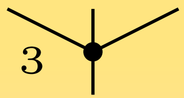
# An example: Boltzmann for trees

Recall that $\mathcal{A}$ is the familly of ordered trees: a tree decomposes into a root and a sequence of subtrees attached by edges:

$$\mathcal{A} = \{\textsf{root}\} \times \mathrm{Seq}(\{\textsf{edge}\} \times \mathcal{A})$$

$$\Gamma[\mathcal{A}](x) := \ \texttt{let}\, k = |\Gamma[\textsf{Seq}](xA(x))|\ \texttt{in}\ \Big(\textsf{root}; (\{\textsf{edge}\} \times \Gamma[\mathcal{A}](x))^k\Big)$$

where the size of a random sequence under the Boltzmann model simply follows a geometric law: $\Pr(|\Gamma[\textsf{Seq}](p)| = k) = p^k(1-p)$.

$\Gamma[\textsf{Seq}] = 3$ 

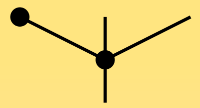
# An example: Boltzmann for trees

Recall that $\mathcal{A}$ is the familly of ordered trees: a tree decomposes into a root and a sequence of subtrees attached by edges:

$$\mathcal{A} = \{\text{root}\} \times \mathrm{Seq}(\{\text{edge}\} \times \mathcal{A})$$

$$\Gamma[\mathcal{A}](x) := \ \texttt{let}\, k = |\Gamma[\mathsf{Seq}](xA(x))| \ \texttt{in}\ \big(\text{root}; (\{\text{edge}\} \times \Gamma[\mathcal{A}](x))^k\big)$$

where the size of a random sequence under the Boltzmann model simply follows a geometric law: $\Pr(|\Gamma[\mathsf{Seq}](p)| = k) = p^k(1-p)$.



# An example: Boltzmann for trees

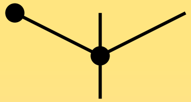
Recall that $\mathcal{A}$ is the familly of ordered trees: a tree decomposes into a root and a sequence of subtrees attached by edges:

$$\mathcal{A} = \{\mathsf{root}\} \times \mathrm{Seq}(\{\mathsf{edge}\} \times \mathcal{A})$$

$$\Gamma[\mathcal{A}](x) := \ \texttt{let}\, k = |\Gamma[\mathsf{Seq}](xA(x))|\ \texttt{in}\ \big(\mathsf{root}; (\{\mathsf{edge}\} \times \Gamma[\mathcal{A}](x))^k\big)$$

where the size of a random sequence under the Boltzmann model simply follows a geometric law: $\Pr(|\Gamma[\mathsf{Seq}](p)| = k) = p^k(1-p)$.

$\Gamma[\mathsf{Seq}] = 2$



# An example: Boltzmann for trees

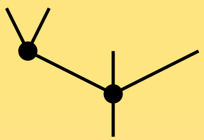
Recall that $\mathcal{A}$ is the familly of ordered trees: a tree decomposes into a root and a sequence of subtrees attached by edges:

$$\mathcal{A} = \{\text{root}\} \times \mathrm{Seq}(\{\text{edge}\} \times \mathcal{A})$$

$$\Gamma[\mathcal{A}](x) := \ \texttt{let}\, k = |\Gamma[\mathsf{Seq}](xA(x))| \ \texttt{in} \ \big(\text{root}; (\{\text{edge}\} \times \Gamma[\mathcal{A}](x))^k\big)$$

where the size of a random sequence under the Boltzmann model simply follows a geometric law: $\Pr(|\Gamma[\mathsf{Seq}](p)| = k) = p^k(1-p)$.

$\Gamma[\mathsf{Seq}] = 2$



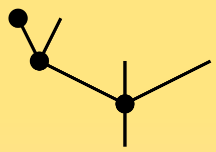
# An example: Boltzmann for trees

Recall that $\mathcal{A}$ is the familly of ordered trees: a tree decomposes into a root and a sequence of subtrees attached by edges:

$$\mathcal{A} = \{\textsf{root}\} \times \mathrm{Seq}(\{\textsf{edge}\} \times \mathcal{A})$$

$$\Gamma[\mathcal{A}](x) := \texttt{let}\, k = |\Gamma[\textsf{Seq}](xA(x))| \;\texttt{in}\; \big(\textsf{root}; (\{\textsf{edge}\} \times \Gamma[\mathcal{A}](x))^k\big)$$

where the size of a random sequence under the Boltzmann model simply follows a geometric law: $\Pr(|\Gamma[\textsf{Seq}](p)| = k) = p^k(1-p)$.



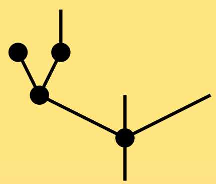
# An example: Boltzmann for trees

Recall that $\mathcal{A}$ is the familly of ordered trees: a tree decomposes into a root and a sequence of subtrees attached by edges:

$$\mathcal{A} = \{\textsf{root}\} \times \mathrm{Seq}(\{\textsf{edge}\} \times \mathcal{A})$$

$$\Gamma[\mathcal{A}](x) := \ \texttt{let}\, k = |\Gamma[\textsf{Seq}](xA(x))| \ \texttt{in} \ \big(\textsf{root}; (\{\textsf{edge}\} \times \Gamma[\mathcal{A}](x))^k\big)$$

where the size of a random sequence under the Boltzmann model simply follows a geometric law: $\Pr(|\Gamma[\textsf{Seq}](p)| = k) = p^k(1-p)$.



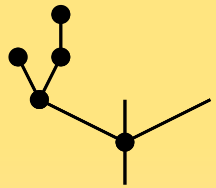
# An example: Boltzmann for trees

Recall that $\mathcal{A}$ is the familly of ordered trees: a tree decomposes into a root and a sequence of subtrees attached by edges:

$$\mathcal{A} = \{\textsf{root}\} \times \mathrm{Seq}(\{\textsf{edge}\} \times \mathcal{A})$$

$$\Gamma[\mathcal{A}](x) := \texttt{let}\, k = |\Gamma[\textsf{Seq}](xA(x))|\ \texttt{in}\ \big(\textsf{root}; (\{\textsf{edge}\} \times \Gamma[\mathcal{A}](x))^k\big)$$

where the size of a random sequence under the Boltzmann model simply follows a geometric law: $\mathrm{Pr}(|\Gamma[\textsf{Seq}](p)| = k) = p^k(1-p)$.



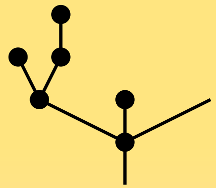
## An example: Boltzmann for trees

Recall that $\mathcal{A}$ is the familly of ordered trees: a tree decomposes into a root and a sequence of subtrees attached by edges:

$$\mathcal{A} = \{\text{root}\} \times \mathrm{Seq}(\{\text{edge}\} \times \mathcal{A})$$

$$\Gamma[\mathcal{A}](x) := \ \texttt{let}\, k = |\Gamma[\mathsf{Seq}](xA(x))| \ \texttt{in}\ \big(\text{root}; (\{\text{edge}\} \times \Gamma[\mathcal{A}](x))^k\big)$$

where the size of a random sequence under the Boltzmann model simply follows a geometric law: $\Pr(|\Gamma[\mathsf{Seq}](p)| = k) = p^k(1-p)$.



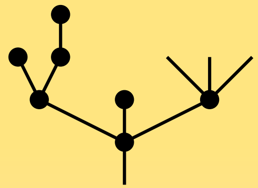
# An example: Boltzmann for trees

Recall that $\mathcal{A}$ is the familly of ordered trees: a tree decomposes into a root and a sequence of subtrees attached by edges:

$$\mathcal{A} = \{\textsf{root}\} \times \mathrm{Seq}(\{\textsf{edge}\} \times \mathcal{A})$$

$$\Gamma[\mathcal{A}](x) := \texttt{ let}\, k = |\Gamma[\textsf{Seq}](xA(x))| \texttt{ in } \left(\textsf{root}; (\{\textsf{edge}\} \times \Gamma[\mathcal{A}](x))^k\right)$$

where the size of a random sequence under the Boltzmann model simply follows a geometric law: $\Pr(|\Gamma[\textsf{Seq}](p)| = k) = p^k(1-p)$.



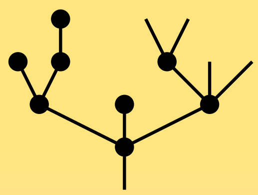
# An example: Boltzmann for trees

Recall that $\mathcal{A}$ is the familly of ordered trees: a tree decomposes into a root and a sequence of subtrees attached by edges:

$$\mathcal{A} = \{\textsf{root}\} \times \mathrm{Seq}(\{\textsf{edge}\} \times \mathcal{A})$$

$$\Gamma[\mathcal{A}](x) := \texttt{ let}\, k = |\Gamma[\textsf{Seq}](xA(x))| \texttt{ in } \big(\textsf{root}; (\{\textsf{edge}\} \times \Gamma[\mathcal{A}](x))^k\big)$$

where the size of a random sequence under the Boltzmann model simply follows a geometric law: $\Pr(|\Gamma[\textsf{Seq}](p)| = k) = p^k(1-p)$.



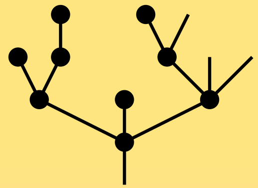
# An example: Boltzmann for trees

Recall that $\mathcal{A}$ is the familly of ordered trees: a tree decomposes into a root and a sequence of subtrees attached by edges:

$$\mathcal{A} = \{\mathsf{root}\} \times \mathrm{Seq}(\{\mathsf{edge}\} \times \mathcal{A})$$

$$\Gamma[\mathcal{A}](x) := \texttt{ let}\, k = |\Gamma[\mathsf{Seq}](xA(x))| \texttt{ in } \left(\mathsf{root}; (\{\mathsf{edge}\} \times \Gamma[\mathcal{A}](x))^k\right)$$

where the size of a random sequence under the Boltzmann model simply follows a geometric law: $\Pr(|\Gamma[\mathsf{Seq}](p)| = k) = p^k(1-p)$.



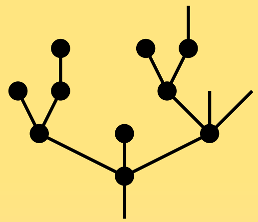
# An example: Boltzmann for trees

Recall that $\mathcal{A}$ is the familly of ordered trees: a tree decomposes into a root and a sequence of subtrees attached by edges:

$$\mathcal{A} = \{\textsf{root}\} \times \mathrm{Seq}(\{\textsf{edge}\} \times \mathcal{A})$$

$$\Gamma[\mathcal{A}](x) := \ \texttt{let}\, k = |\Gamma[\textsf{Seq}](xA(x))|\ \texttt{in}\ \big(\textsf{root}; (\{\textsf{edge}\} \times \Gamma[\mathcal{A}](x))^k\big)$$

where the size of a random sequence under the Boltzmann model simply follows a geometric law: $\Pr(|\Gamma[\textsf{Seq}](p)| = k) = p^k(1-p)$.



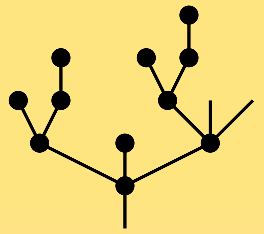
# An example: Boltzmann for trees

Recall that $\mathcal{A}$ is the familly of ordered trees: a tree decomposes into a root and a sequence of subtrees attached by edges:

$$\mathcal{A} = \{\textsf{root}\} \times \mathrm{Seq}(\{\textsf{edge}\} \times \mathcal{A})$$

$$\Gamma[\mathcal{A}](x) := \texttt{ let}\, k = |\Gamma[\textsf{Seq}](xA(x))| \texttt{ in } \big(\textsf{root}; (\{\textsf{edge}\} \times \Gamma[\mathcal{A}](x))^k\big)$$

where the size of a random sequence under the Boltzmann model simply follows a geometric law: $\mathrm{Pr}(|\Gamma[\textsf{Seq}](p)| = k) = p^k(1-p)$.



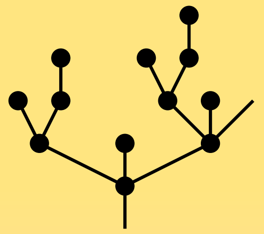
# An example: Boltzmann for trees

Recall that $\mathcal{A}$ is the familly of ordered trees: a tree decomposes into a root and a sequence of subtrees attached by edges:

$$\mathcal{A} = \{\mathsf{root}\} \times \mathrm{Seq}(\{\mathsf{edge}\} \times \mathcal{A})$$

$$\Gamma[\mathcal{A}](x) := \texttt{ let}\, k = |\Gamma[\mathsf{Seq}](xA(x))| \texttt{ in } \big(\mathsf{root}; (\{\mathsf{edge}\} \times \Gamma[\mathcal{A}](x))^k\big)$$

where the size of a random sequence under the Boltzmann model simply follows a geometric law: $\Pr(|\Gamma[\mathsf{Seq}](p)| = k) = p^k(1-p)$.



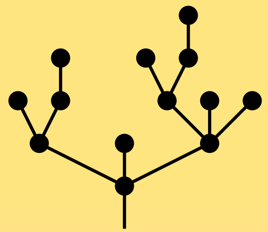
# An example: Boltzmann for trees

Recall that $\mathcal{A}$ is the familly of ordered trees: a tree decomposes into a root and a sequence of subtrees attached by edges:

$$\mathcal{A} = \{\mathsf{root}\} \times \mathrm{Seq}(\{\mathsf{edge}\} \times \mathcal{A})$$

$$\Gamma[\mathcal{A}](x) := \ \texttt{let}\, k = |\Gamma[\mathsf{Seq}](xA(x))| \ \texttt{in} \ \big(\mathsf{root}; (\{\mathsf{edge}\} \times \Gamma[\mathcal{A}](x))^k\big)$$

where the size of a random sequence under the Boltzmann model simply follows a geometric law: $\Pr(|\Gamma[\mathsf{Seq}](p)| = k) = p^k(1-p)$.



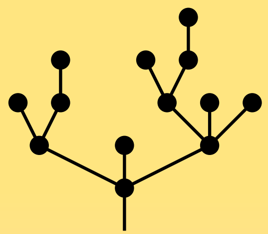
# An example: Boltzmann for trees

Recall that $\mathcal{A}$ is the familly of ordered trees: a tree decomposes into a root and a sequence of subtrees attached by edges:

$$\mathcal{A} = \{\text{root}\} \times \mathrm{Seq}(\{\text{edge}\} \times \mathcal{A})$$

$$\Gamma[\mathcal{A}](x) := \texttt{ let } k = |\Gamma[\text{Seq}](xA(x))| \texttt{ in } \big(\text{root}; (\{\text{edge}\} \times \Gamma[\mathcal{A}](x))^k\big)$$

where the size of a random sequence under the Boltzmann model simply follows a geometric law: $\Pr(|\Gamma[\text{Seq}](p)| = k) = p^k(1-p)$.



The generation finishes with proba 1.

The probability to get size $n$ depends on the choice of $x$, increasing near the singularity: if $x_n = \frac{1}{4}(1 - \frac{1}{n})$

$$\Pr\left(|\Gamma[\mathcal{A}](x_n))| = n\right) = \frac{|\mathcal{A}_n| \cdot x^n}{A(x)} \approx 4^n n^{-3/2} \left(\tfrac{1}{4}(1 - \tfrac{1}{n})\right)^n \approx n^{-3/2}$$

The expected size of a Boltzmann tree of parameter $x_n = \frac{1}{4}(1 - \frac{1}{n})$ is $\mathbb{E}(|\Gamma[\mathcal{A}](x_n)|) = \frac{A(x_n)'}{A(x_n)} \approx (1 - 4x_n)^{-1/2} = \sqrt{n}$

# Uniform sampling via Boltzmann

The probability to get $\Gamma[\mathcal{A}](x) = a$ depends only on the size of $a$.

Hence the uniform random generator:

$\mathsf{U}[\mathcal{A}(n)] :=$ `do let` $a = \Gamma[\mathcal{A}](x)$ **`until`** $|a| = n$; **`return`** $a$;

# Boltzmann in progress

Initial model: Labelled and rigid unlabelled structures

*Duchon, Flajolet, Louchard, Schaeffer (2002)*

Unlabelled structures and Polya theory

*Flajolet, Fusy, Pivoteau (2007)* and *Bodirsky, Fusy, Kang and Vigerske (2007)*

Efficient oracles for the evaluation of generating series

*Pivoteau, Salvy, Soria (2008)*

Graphs properties via Boltzmann models

*Bernasconi, Panagiotou, Steger, Weißt (2006)*

Complex structures: Apollonian structures, XML documents

*Darasse, Soria (2007), Darasse (2008)*

Complex structures: plane partitions, colored objects

*Bodini, Fusy, Pivoteau (2006), Bodini, Jacquot (2008)*

Complex structures: deterministic automata

*Bassino, NIcaud (2006), Bassino, David, Nicaud (2008)*

Complex structures: planar graphs

*Fusy (2006)*

# Planar graphs, planar maps, and surfaces

A planar graph: there exists an embedding in the plane

A planar map: the (combinatorial) embedding in the plane is fixed

# Planar graphs, planar maps, and surfaces

A planar graph: there exists an embedding in the ~~plane~~
sphere

A planar map: the (combinatorial) embedding in the ~~plane~~ is fixed
sphere

Surfaces: let $\mathcal{S}_g$ be the compact orientable surface of genus $g$: $\mathcal{S}_0$ is the sphere, $\mathcal{S}_1$ the torus; in general $\mathcal{S}_g$ is a "sphere" with $g$ handles.

A map of genus $g$: an embedding of a graph on $\mathcal{S}_g$ (faces must be simply connected): Euler's formula: $v + f = e + 2 - 2g$.

A graph of genus $g$: $g$ is the minimum genus of a surface on which the graph can be embedded.

# Random planar maps

Maps are somewhat easier to deal with.

Start with maps

1-c planar maps = Closure(well labelled ordered trees)

2-c planar maps = Core(1-c planar maps)

3-c planar maps = Core(2-c planar maps) = Closure (binary trees)

3Core(2Core(Closure($\Gamma[\mathcal{A}^3](x)$))) is Boltzmann

# Random planar graphs (rough idea of Eric Fusy's algorithm)

10 steps to planar graphs (title from Liskovets and Walsh, 87)
Decomposition for planar graphs have been available from decades: the equations were partially written several times until the asymptotic was done by Gimenez and Noy, and efficient random generation by Fusy

labelled planar graphs = sets of 1-connected planar graphs

rooted 1-c planar graphs = (2-c planar graphs)$\circ$(1-c planar graphs)

rooted 2-c planar graphs = (3-c planar graphs)$\circ$(2-c planar graphs)

3-c planar graphs = 3-c planar maps

# Random planar graphs (rough idea of Eric Fusy's algorithm)

10 steps to planar graphs (title from Liskovets and Walsh, 87)
Decomposition for planar graphs have been available from decades: the equations were partially written several times until the asymptotic was done by Gimenez and Noy, and efficient random generation by Fusy

labelled planar graphs = sets of 1-connected planar graphs

rooted 1-c planar graphs = (2-c planar graphs)$\circ$(1-c planar graphs)

rooted 2-c planar graphs = (3-c planar graphs)$\circ$(2-c planar graphs)

3-c planar graphs = 3-c planar maps

Illustration:

# Random planar graphs and maps: some remarkable properties

the decomposition tree: iterate the decomposition

the decomposition is "symmetric": the starting point does not matter

there is a unique giant node in the tree

Bender-Richmond-Wormald, Gao-Wormald, Banderier-Flajolet-Schaeffer-Soria, Gimenez-Noy, Panagiotou-Stenger

the distance between two vertices is of order $n^{1/4}$

the graph/map/giant component converge to the continuum brown

Recall that *emphasised statements* are conjectures

## Random graphs on surfaces

the same picture remains true "almost surely":

the genus is a.s. concentrated in the giant component

proved for maps (Chapuy, Kang, Schaeffer), not yet for graphs

Uniform on the set of graphs that can be embedded in $\mathcal{S}_g$:

have a.s. minimum genus $g$, concentrated in one 3-connected component with unique embedding

at fixed genus $g$, distances are of order $n^{1/4}$ and the limit is the continuum random map of genus $g$.

The problem (should) boils down to sampling maps of genus $g$

Recall that *emphasised statements* are conjectures

# Random graphs on surfaces

Recall that *emphasised statements* are conjectures

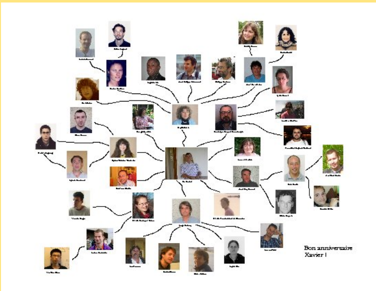
# Combinatorics

# Combinatorics

Combinatorial objects

# Combinatorics

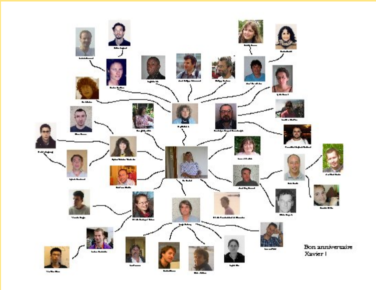
Combinatorial objects

tree like structures

# Combinatorics

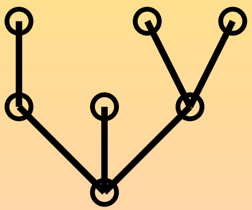
Combinatorial objects





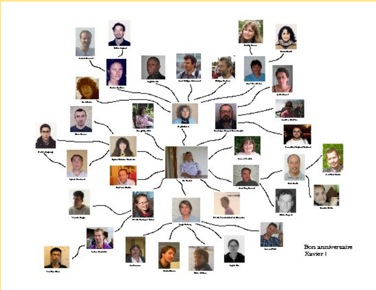
tree like structures

concept of graph



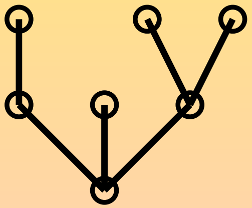
# Combinatorics

## Combinatorial objects

tree like structures

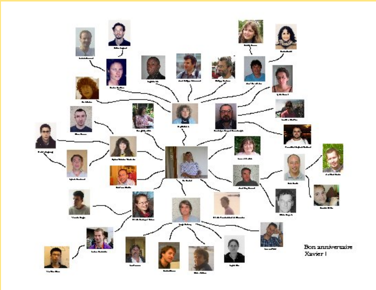
~~concept of graph~~



concept of tree

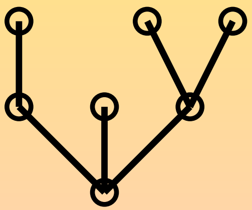
# Combinatorics

## Combinatorial objects

tree like structures

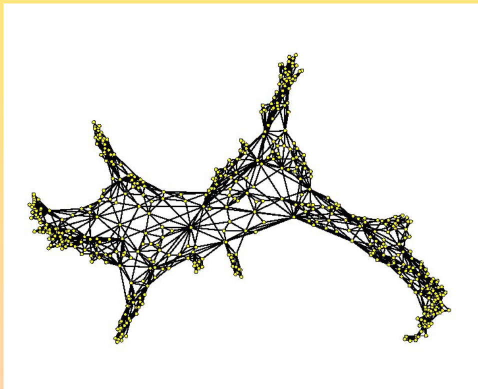
~~concept of graph~~

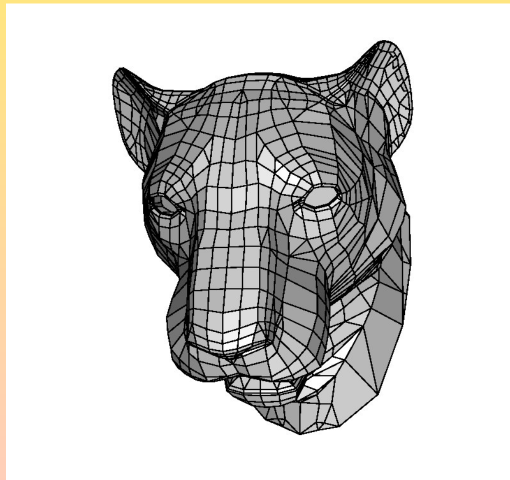


concept of tree

2d discrete structures

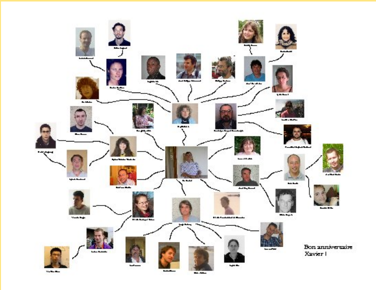
(discretized surfaces, meshes,...)



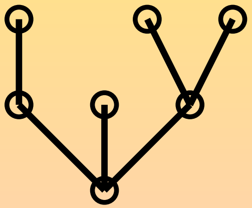
# Combinatorics

## Combinatorial objects

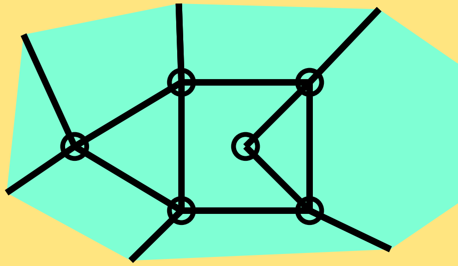




tree like structures

~~concept of graph~~



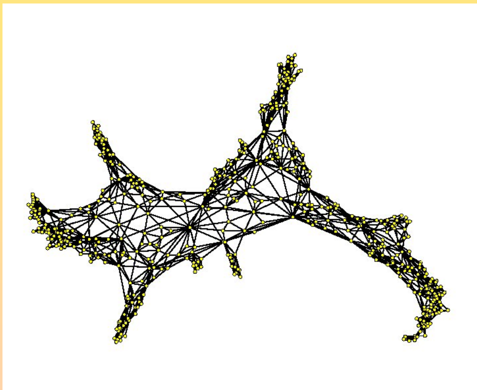
concept of tree

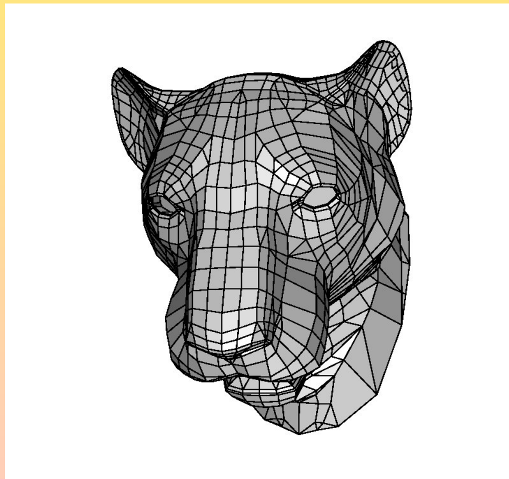


concept of graph

2d discrete structures

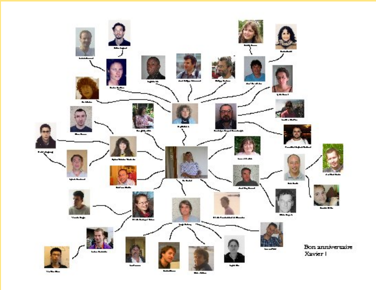
(discretized surfaces, meshes,...)





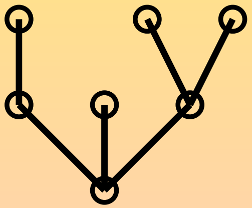
# Combinatorics

## Combinatorial objects

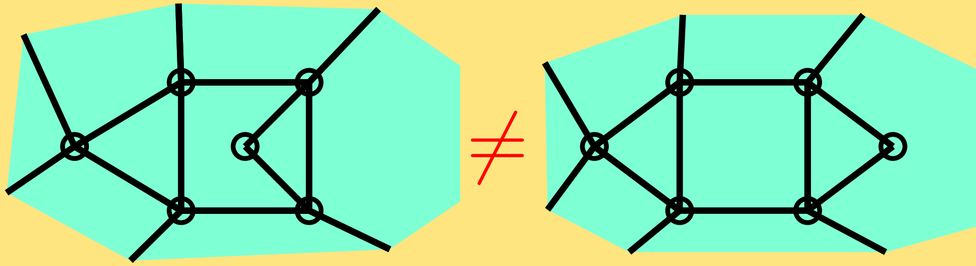

tree like structures

~~concept of graph~~



concept of tree

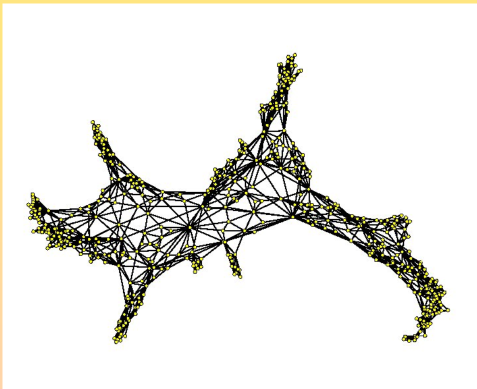
concept of *map*

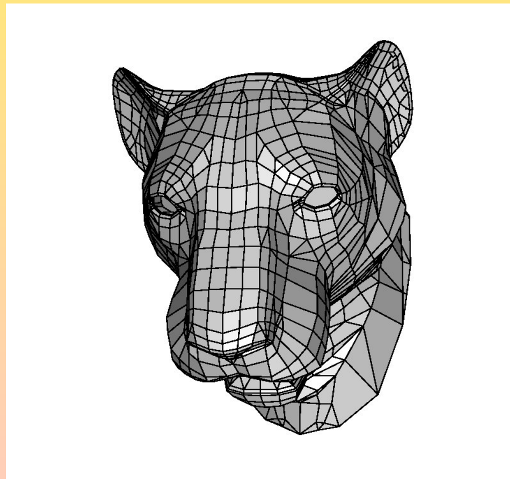


~~concept of graph~~

2d discrete structures

(discretized surfaces, meshes,...)

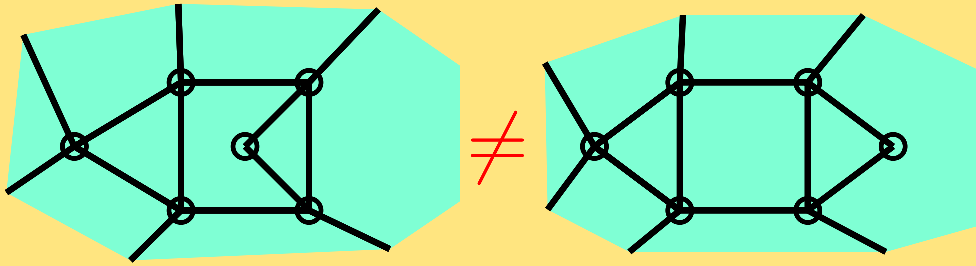




# Combinatorics

Combinatorial objects = discrete abstractions of fundamental structures

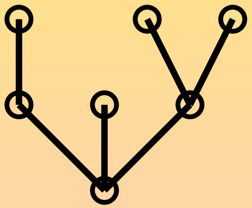
concept of *map*



~~concept of graph~~

tree like structures

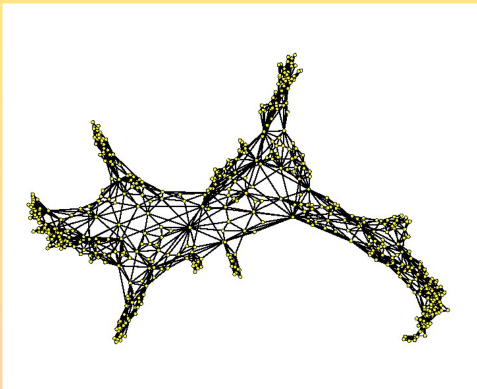
~~concept of graph~~

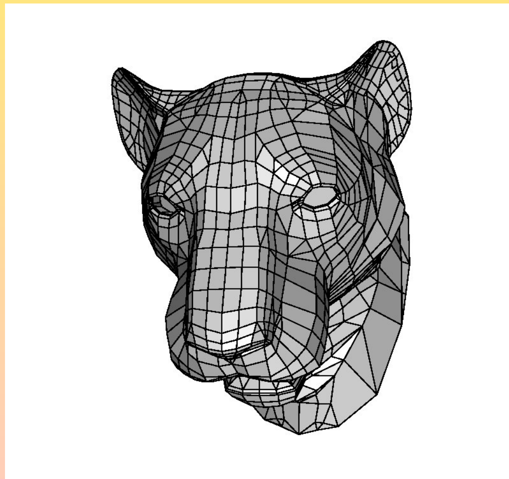


concept of tree

2d discrete structures

(discretized surfaces, meshes,...)





## My idea of combinatorics

Elucidate the properties of those fundamental discrete structures that are common to various scientific fields (CS/math/physics/bio).

# Algorithmic combinatorics

## My idea of combinatorics

Elucidate the properties of those fundamental discrete structures that are common to various scientific fields (CS/math/physics/bio).

## and, more specifically of "algorithmic combinatorics"

concentrate on constructive properties and on the algorithmic point of view on structures

# Algorithmic combinatorics

## My idea of combinatorics

Elucidate the properties of those fundamental discrete structures that are common to various scientific fields (CS/math/physics/bio).

## and, more specifically of "algorithmic combinatorics"

concentrate on constructive properties and on the algorithmic point of view on structures

## The example of trees...

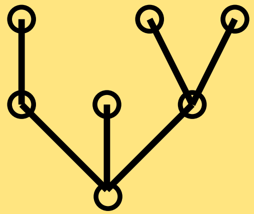
mathematical pt of view: connected graphs without cycle

algorithmic pt of view: recursive description (root; subtrees)
$\Rightarrow$ concept of breadth first or depth first search,
links with context free languages

(... Schützenberger's methodology...)

# Exploration algorithms

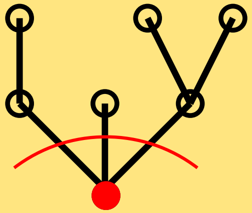
## Tree exploration



breadth first

# Exploration algorithms

## Tree exploration

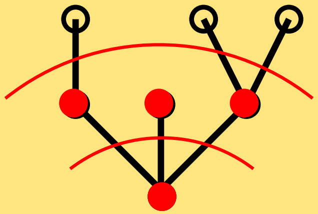


breadth first

# Exploration algorithms

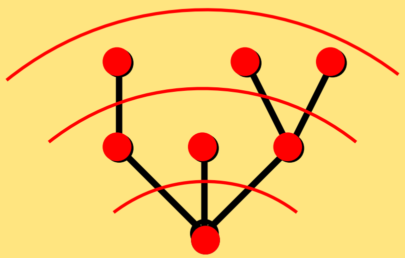
## Tree exploration

breadth first



# Exploration algorithms

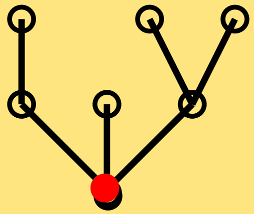
## Tree exploration



breadth first

# Exploration algorithms

## Tree exploration

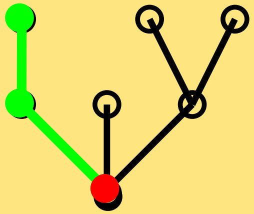


breadth first

depth first

# Exploration algorithms

## Tree exploration

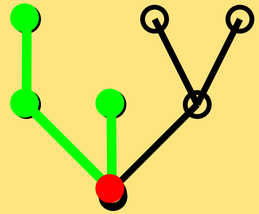


breadth first

depth first

# Exploration algorithms

## Tree exploration

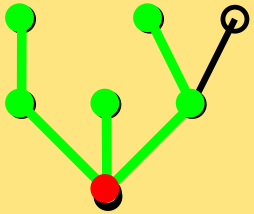


breadth first

depth first

# Exploration algorithms

## Tree exploration

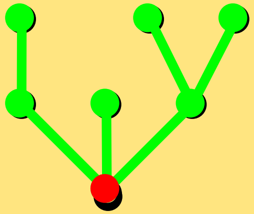


breadth first

depth first

# Exploration algorithms

## Tree exploration

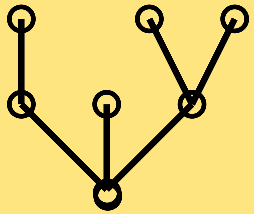


breadth first

depth first

# Exploration algorithms

## Tree exploration



breadth first

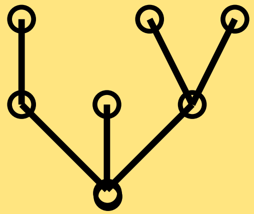
depth first

fundamental tools

for instance to encode trees

# Exploration algorithms

## Tree exploration



breadth first

depth first

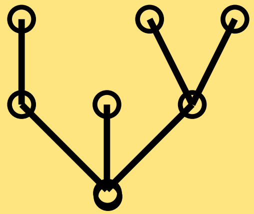
fundamental tools

for instance to encode trees

$\Rightarrow$ the prefix code of a tree

# Exploration algorithms

## Tree exploration



breadth first

depth first

fundamental tools

for instance to encode trees

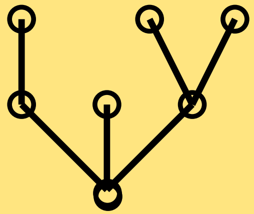
$\Rightarrow$ the prefix code of a tree

3 1 0 2 0 0 0 (breadth first)

3 1 0 0 2 0 0 (depth first)

# Exploration algorithms

## Tree exploration



breadth first

depth first

fundamental tools

for instance to encode trees

$\Rightarrow$ the prefix code of a tree

3 1 0 2 0 0 0 (breadth first)

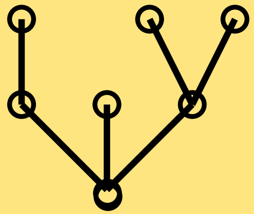
3 1 0 0 2 0 0 (depth first)

**Statement.** The set of code words is easy to describe.

More precisely: the language of prefix codes of ordered trees is *context-free*.

# Exploration algorithms

## Tree exploration



breadth first

depth first

fundamental tools

for instance to encode trees

⇒ the prefix code of a tree

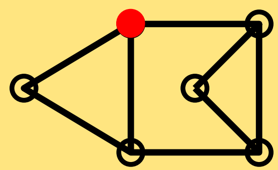
3 1 0 2 0 0 0 (breadth first)

3 1 0 0 2 0 0 (depth first)

**Statement.** The set of code words is easy to describe.

More precisely: the language of prefix codes of ordered trees is *context-free*.

## Graph exploration

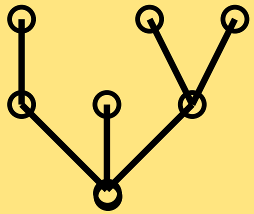


breadth first

depth first

# Exploration algorithms

## Tree exploration



breadth first

depth first

fundamental tools

for instance to encode trees

$\Rightarrow$ the prefix code of a tree

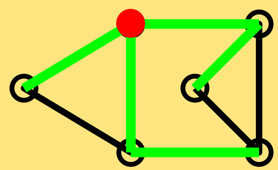
3 1 0 2 0 0 0 (breadth first)

3 1 0 0 2 0 0 (depth first)

**Statement.** The set of code words is easy to describe.

More precisely: the language of prefix codes of ordered trees is *context-free*.

## Graph exploration



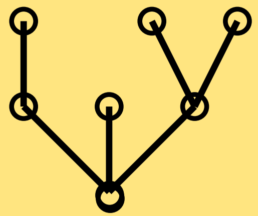
breadth first

depth first

construct a tree along the exploration

# Exploration algorithms

## Tree exploration



breadth first

depth first

fundamental tools

for instance to encode trees

⇒ the prefix code of a tree

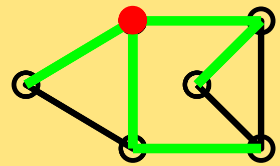
3 1 0 2 0 0 0 (breadth first)

3 1 0 0 2 0 0 (depth first)

**Statement.** The set of code words is easy to describe.

More precisely: the language of prefix codes of ordered trees is *context-free*.

## Graph exploration



breadth first

depth first

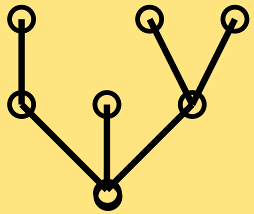
construct a tree along the exploration

$+$ extra info for external edges

⇒ encode graphs by tree-like structures

# Exploration algorithms

## Tree exploration



breadth first

depth first

fundamental tools

for instance to encode trees

$\Rightarrow$ the prefix code of a tree

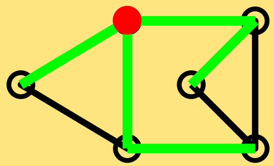
3 1 0 2 0 0 0 (breadth first)

3 1 0 0 2 0 0 (depth first)

**Statement.** The set of code words is easy to describe.

More precisely: the language of prefix codes of ordered trees is *context-free*.

## Graph exploration



breadth first

depth first

construct a tree along the exploration

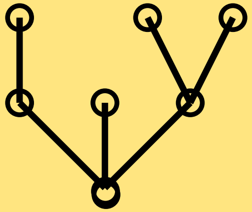
$+$ extra info for external edges

$\Rightarrow$ encode graphs by tree-like structures

but the set of "coding" trees is not easy to describe (for classic families of graphs like planar, 3-connected,...)

# Exploration algorithms

## Tree exploration



breadth first

depth first

fundamental tools

for instance to encode trees

$\Rightarrow$ the prefix code of a tree

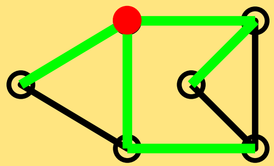
3 1 0 2 0 0 0 (breadth first)

3 1 0 0 2 0 0 (depth first)

**Statement.** The set of code words is easy to describe.

More precisely: the language of prefix codes of ordered trees is *context-free*.

## Graph exploration



breadth first

depth first

construct a tree along the exploration

+ extra info for external edges

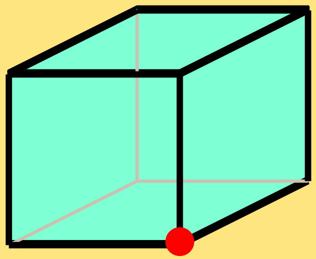
$\Rightarrow$ encode graphs by tree-like structures

but the set of "coding" trees is not easy to describe (for classic families of graphs like planar, 3-connected,...)

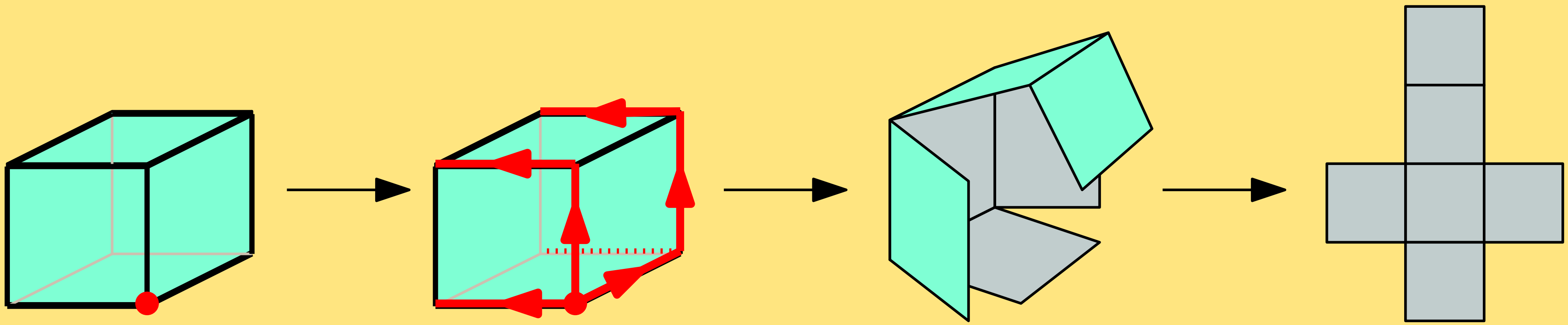
No good analog of the previous "statement".

## Exploration of a map and surface surgery

## Exploration of a map and surface surgery

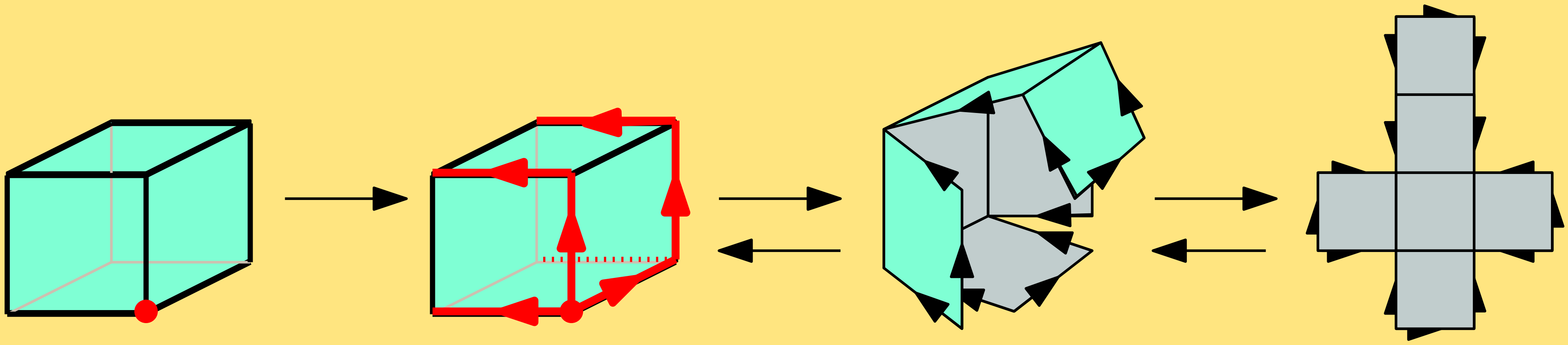


## Exploration of a map and surface surgery



Exploration + cut $\Rightarrow$ a "net" of the map

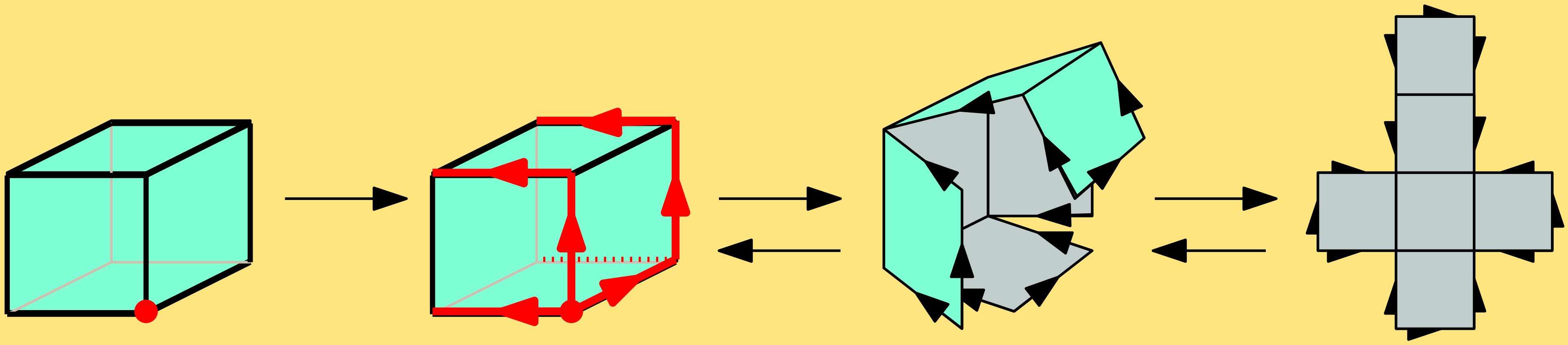
## Exploration of a map and surface surgery



Exploration + cut $\Rightarrow$ a "net" of the map

in order to reconstruct the surface, the orientation of cuts is enough: merge adjacent converging sides + iterate

## Exploration of a map and surface surgery

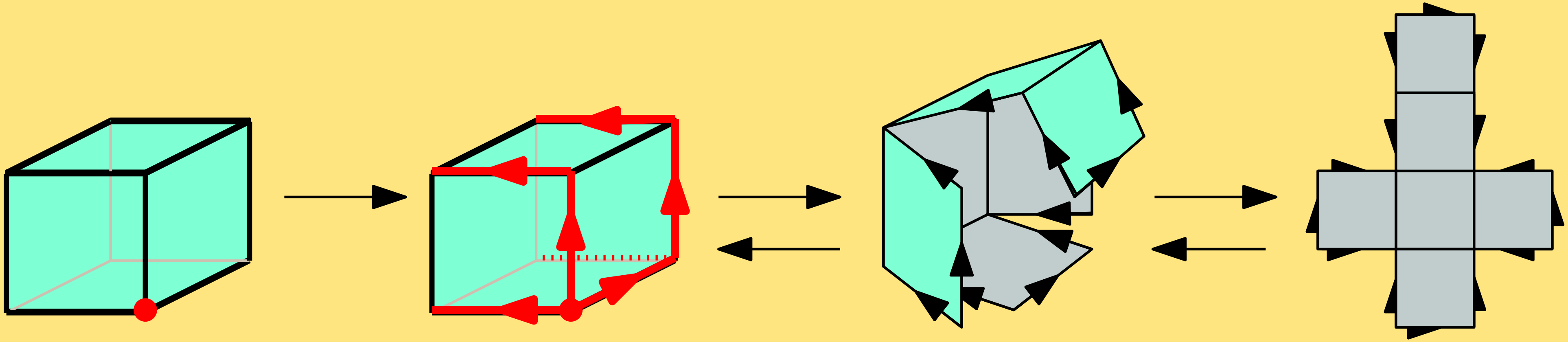


Exploration + cut ⇒ a "net" of the map

in order to reconstruct the surface, the orientation of cuts is enough: merge adjacent converging sides + iterate

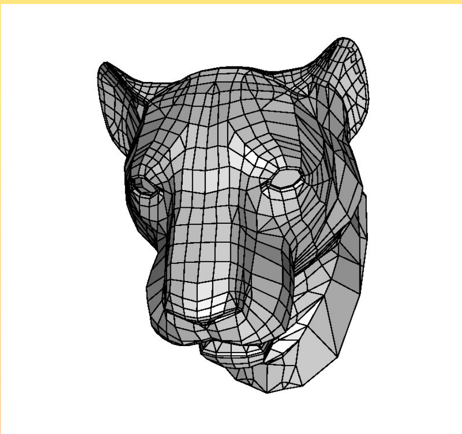
# Exploration algorithms

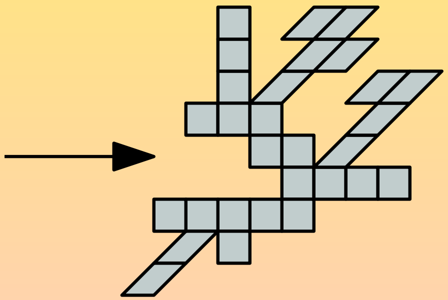
## Exploration of a map and surface surgery



Exploration + cut $\Rightarrow$ a "net" of the map

in order to reconstruct the surface, the orientation of cuts is enough: merge adjacent converging sides + iterate



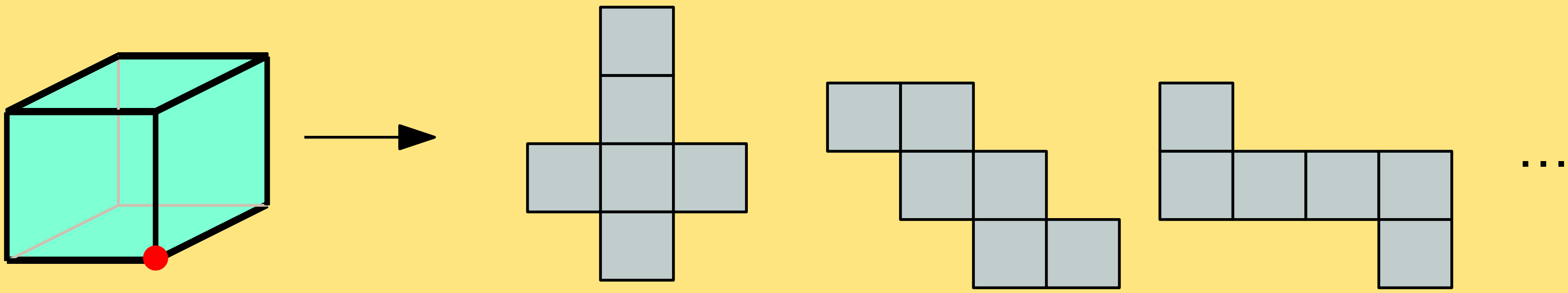


Nets are always trees of polygons

(as long as the surface has no handle)

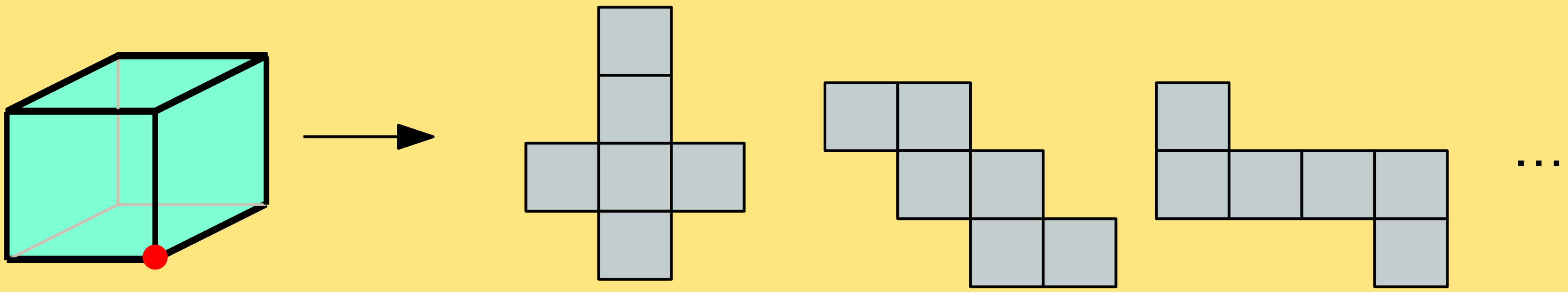
# Exploration algorithms

To a map are associated many different nets

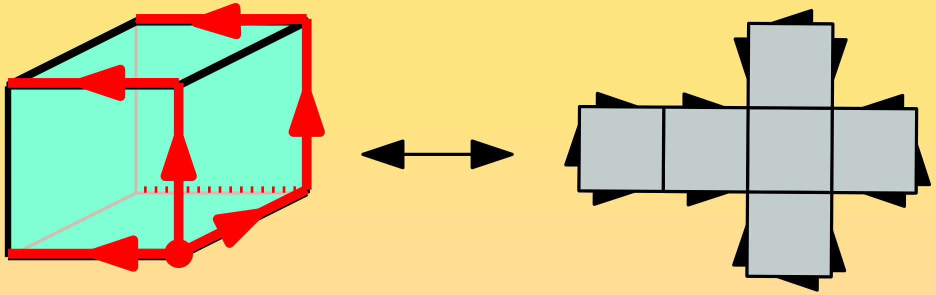


# Exploration algorithms

To a map are associated many different nets

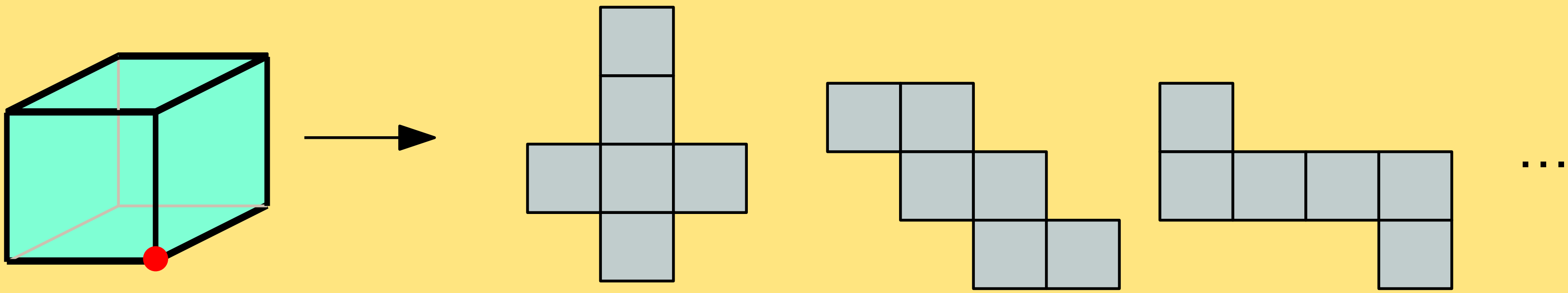


but a given exploration algorithm associates a canonical net to each map

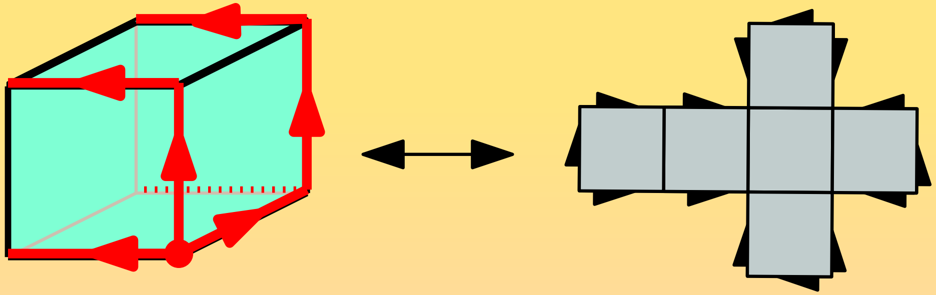


# Exploration algorithms

To a map are associated many different nets



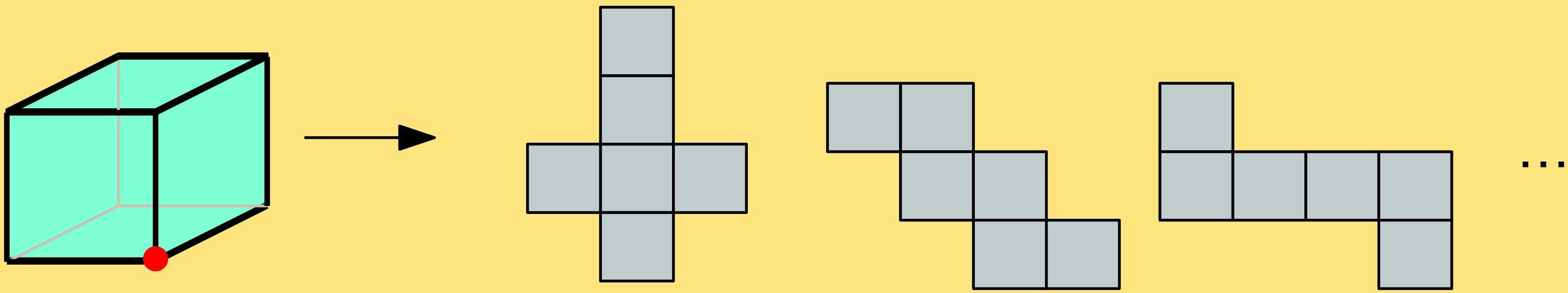
but a given exploration algorithm associates a canonical net to each map



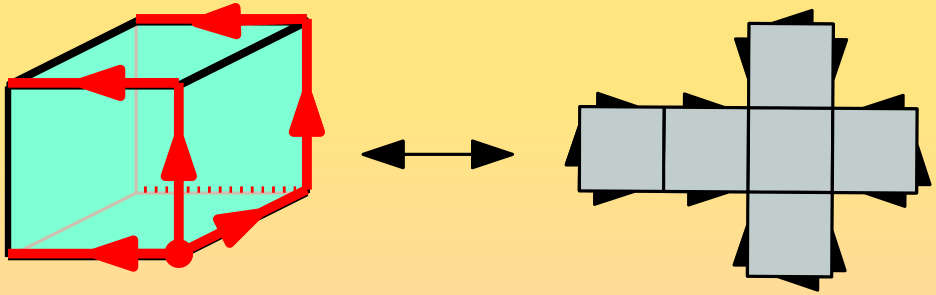
Represent again a map by a tree like structure!

# Exploration algorithms

To a map are associated
many different nets



but a given exploration algorithm associates a canonical net to each map

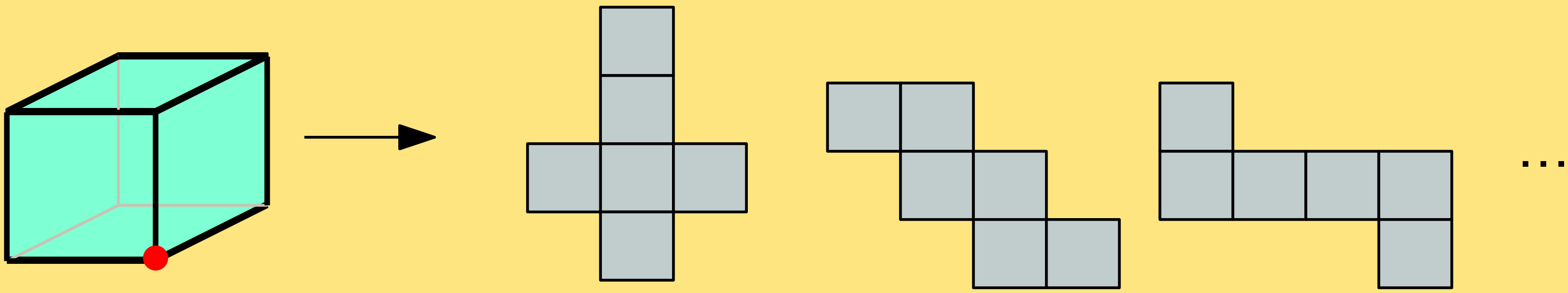


Represent again a map by a tree like structure!

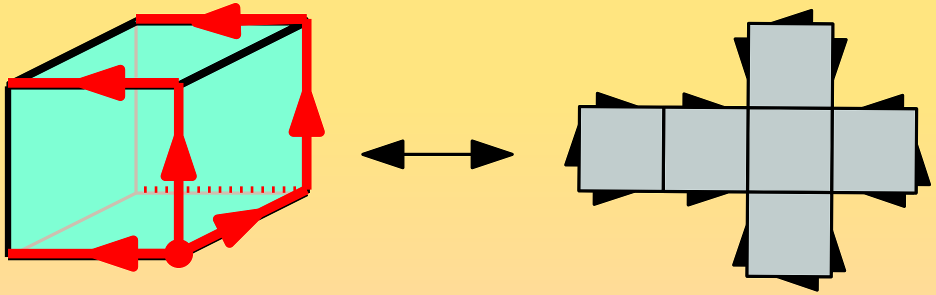
Each exploration algo $\Rightarrow$ a bijection, but what is the set of valid nets?

# Exploration algorithms

To a map are associated
many different nets




but a given exploration algorithm associates a canonical net to each map



Represent again a map by
a tree like structure!

Each exploration algo $\Rightarrow$ a bijection, but what is the set of valid nets?

Valid nets are easier to describe than exploration trees!

## Exploration algorithms

### Statement

To many natural families of maps is associated a standard exploration algorithms (breadth first, depth first, Schnyder,...) such that the cut yields *context-free* nets.

# Exploration algorithms

## Statement

To many natural families of maps is associated a standard exploration algorithms (breadth first, depth first, Schnyder,...) such that the cut yields *context-free* nets.

this statment covers a series of "coherent" theorems

- Cori-Vauquelin 1984, S. 1997, Marcus-S. 1998, Bousquet-Mélou-S. 1999, Poulalhon-S. 2003, Bouttier-di Francesco-Guitter 2004, Fusy-Poulalhon-S. 2005, Bernardi 2006

# Exploration algorithms

## Statement

To many natural families of maps is associated a standard exploration algorithms (breadth first, depth first, Schnyder,...) such that the cut yields *context-free* nets.

### this statment covers a series of "coherent" theorems

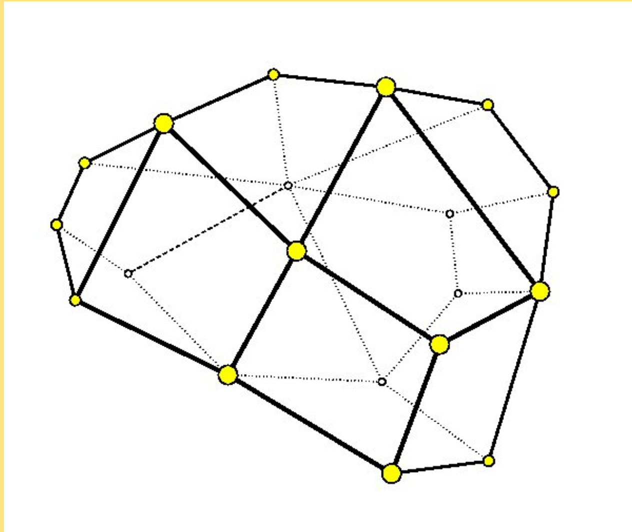
- Cori-Vauquelin 1984, S. 1997, Marcus-S. 1998, Bousquet-Mélou-S. 1999, Poulalhon-S. 2003, Bouttier-di Francesco-Guitter 2004, Fusy-Poulalhon-S. 2005, Bernardi 2006

### with various types of applications

- optimal encodings and compact data structures for meshes
- random sampling and automatic drawing of graph and map
- enumeration: maps, ramified coverings, alternating knots...
- random discrete surfaces

# Application to discrete random surfaces

Planar quadrangulations (quads) as a model of discretized spheres

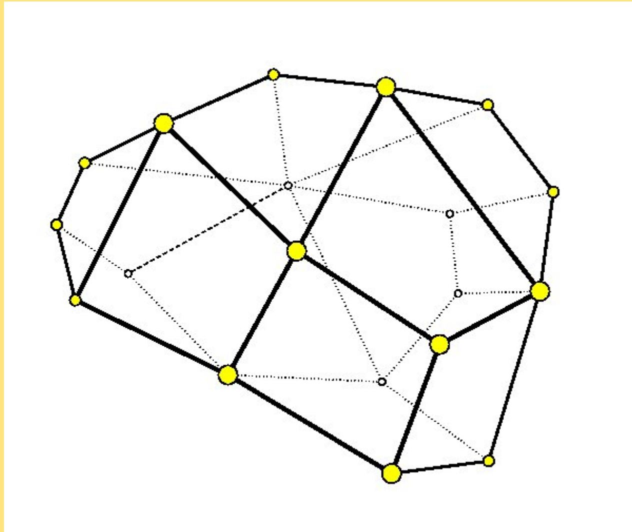


Let $|Q_n|$ be the set of quads with $n$ faces and $X_n$ be a uniform random quad of $Q_n$:

$$\Pr(X_n = q) = \frac{1}{|Q_n|}, \quad \forall q \in Q_n$$

# Application to discrete random surfaces

Planar quadrangulations (quads) as a model of discretized spheres



Let $|Q_n|$ be the set of quads with $n$ faces and $X_n$ be a uniform random quad of $Q_n$:

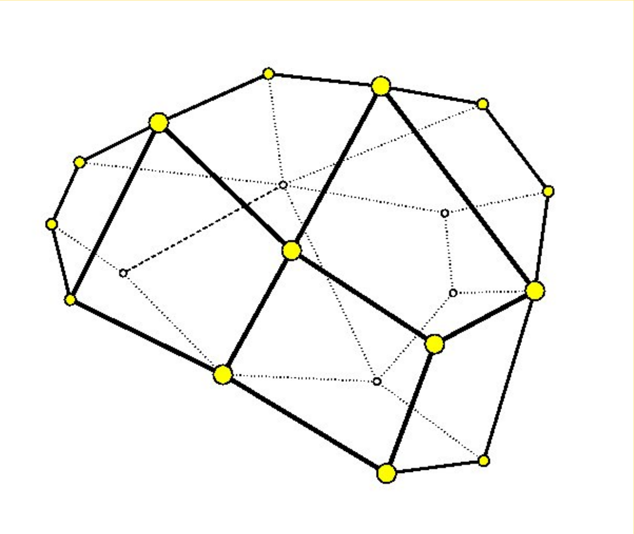
$$\Pr(X_n = q) = \frac{1}{|Q_n|}, \quad \forall q \in Q_n$$

This model of random geometries is called *2d discrete quantum gravity* in statistical $\varphi$.

Lots of results via the celebrated method of topological expansion of matrix integrals (Brezin, Itzykson, Parisi, Zuber, 72).

# Application to discrete random surfaces

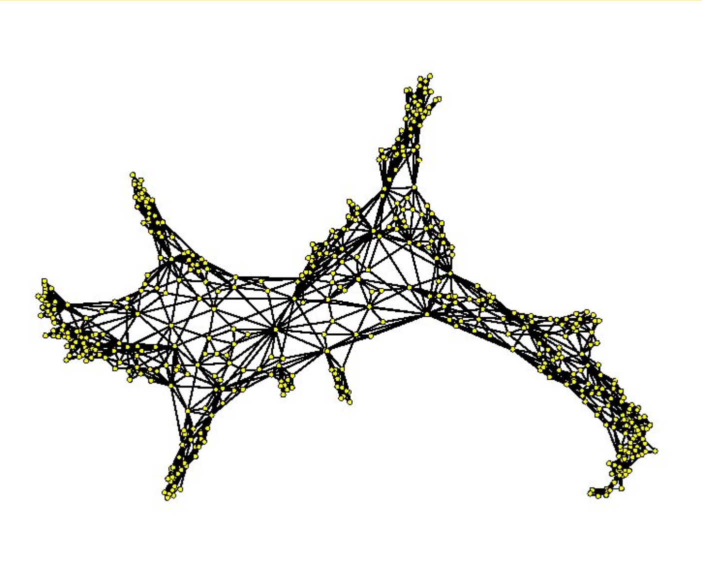
Planar quadrangulations (quads) as a model of discretized spheres



Let $|Q_n|$ be the set of quads with $n$ faces and $X_n$ be a uniform random quad of $Q_n$:

$$\Pr(X_n = q) = \frac{1}{|Q_n|}, \quad \forall q \in Q_n$$

This model of random geometries is called *2d discrete quantum gravity* in statistical $\varphi$.

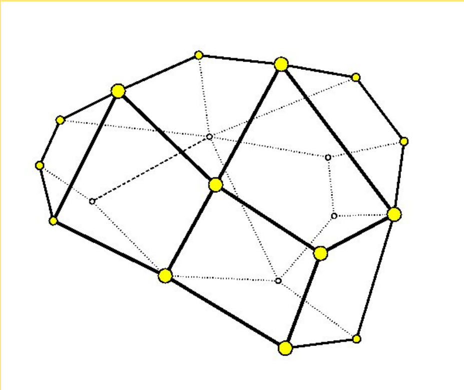


Lots of results via the celebrated method of topological expansion of matrix integrals (Brezin, Itzykson, Parisi, Zuber, 72).

But this approach does not allow to study the intrinsec geometry of these surface!

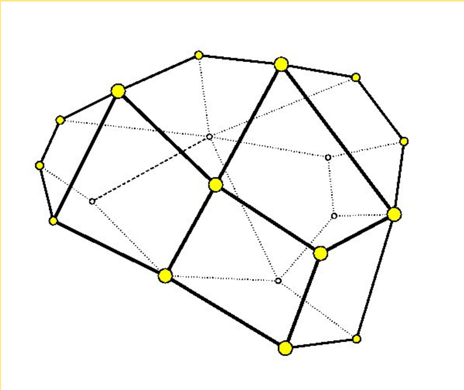
# Quadrangulations via breadth first search

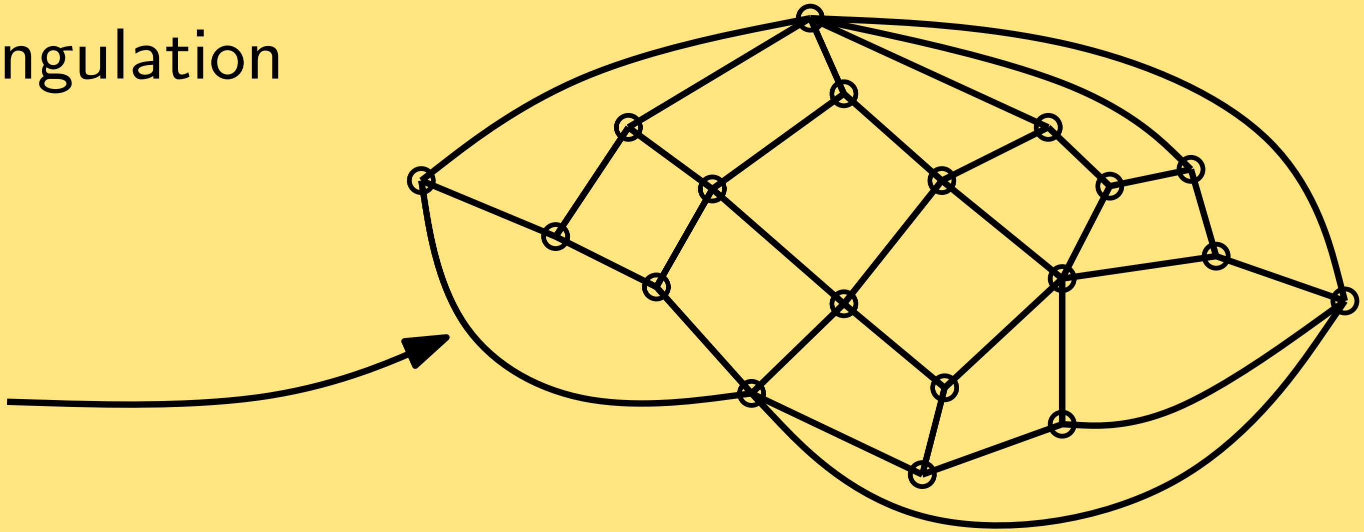
Consider a planar quadrangulation



# Quadrangulations via breadth first search

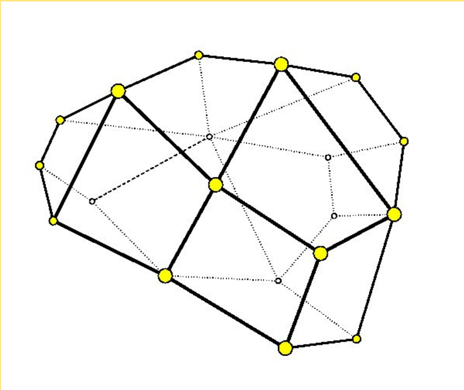
Consider a planar quadrangulation

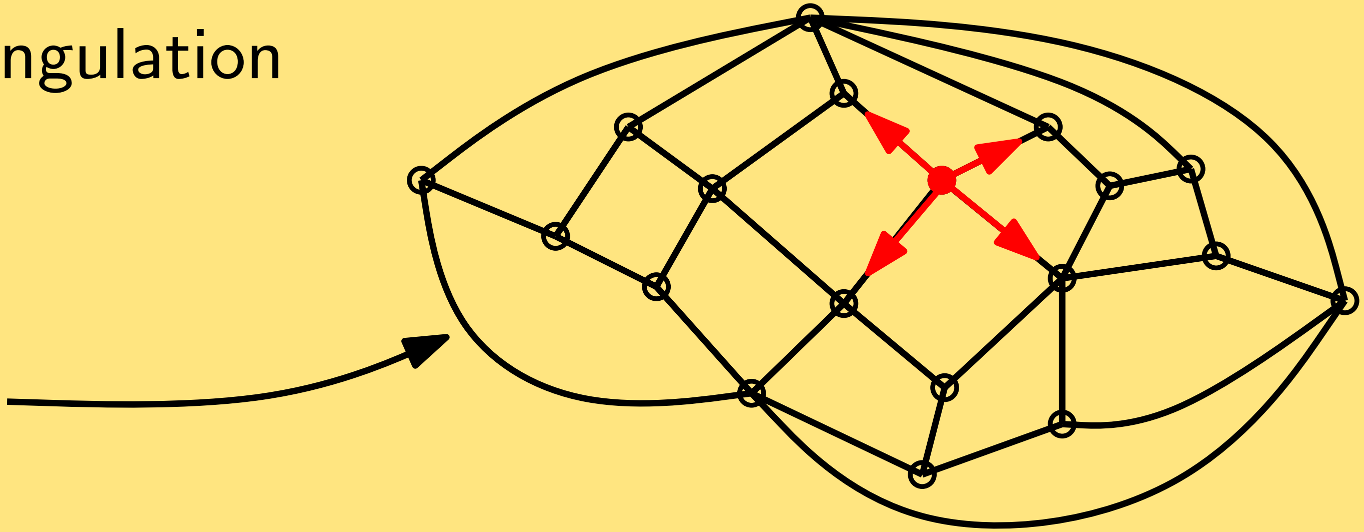




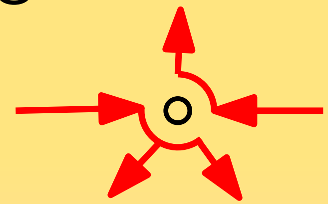
# Quadrangulations via breadth first search

Consider a planar quadrangulation



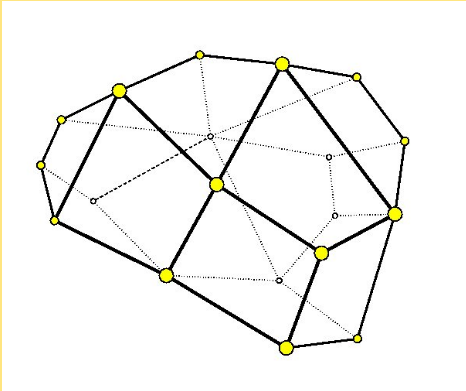


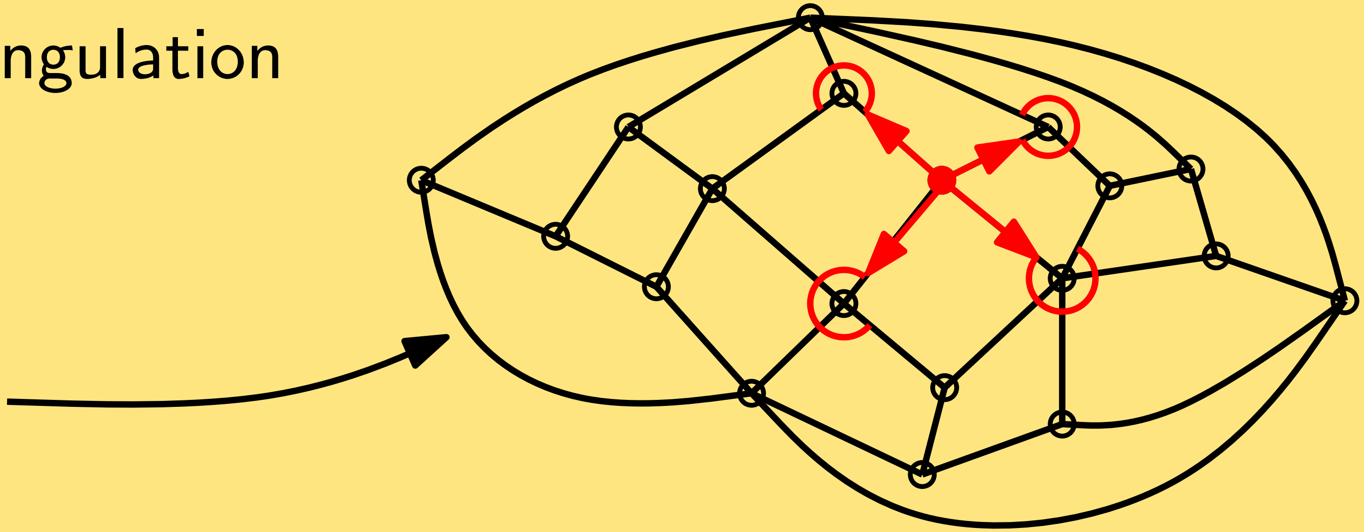
Apply bfs with the rotatoria rule
and cut along the flow



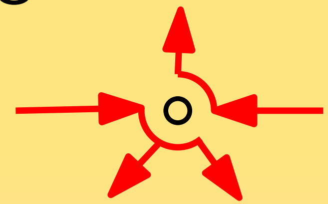
# Quadrangulations via breadth first search

Consider a planar quadrangulation



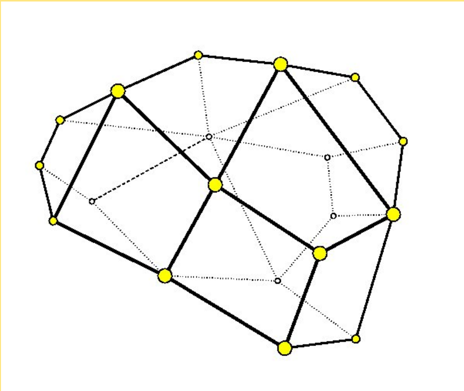


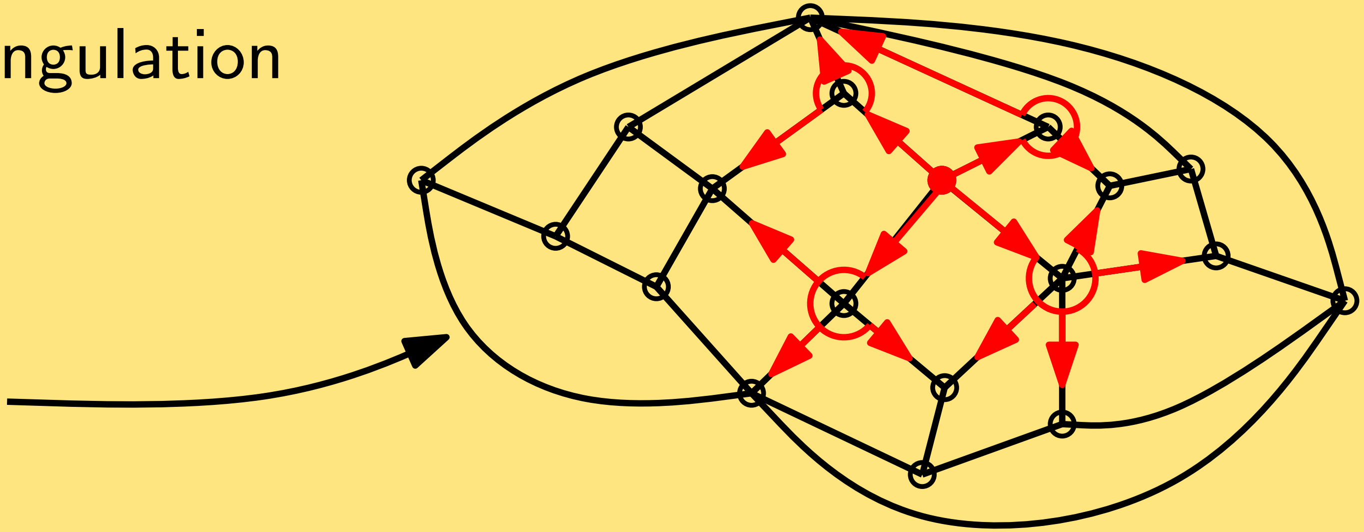
Apply bfs with the rotatoria rule and cut along the flow



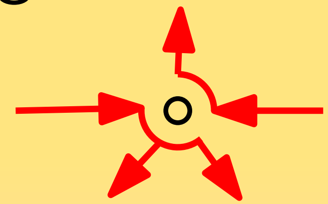
# Quadrangulations via breadth first search

Consider a planar quadrangulation



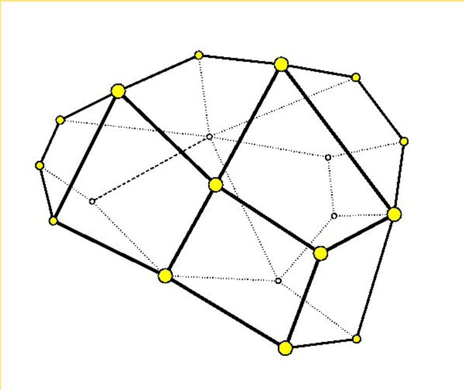


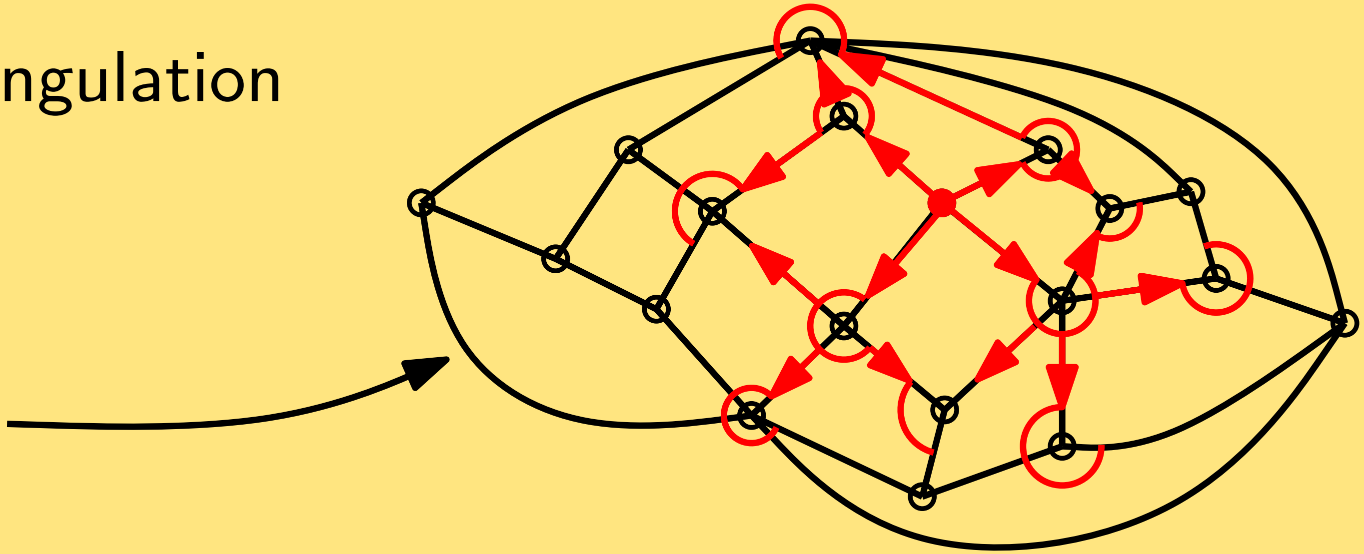
Apply bfs with the rotatoria rule and cut along the flow



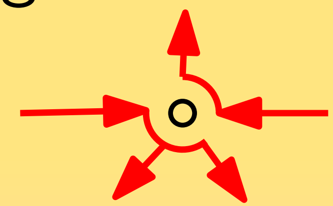
# Quadrangulations via breadth first search

Consider a planar quadrangulation



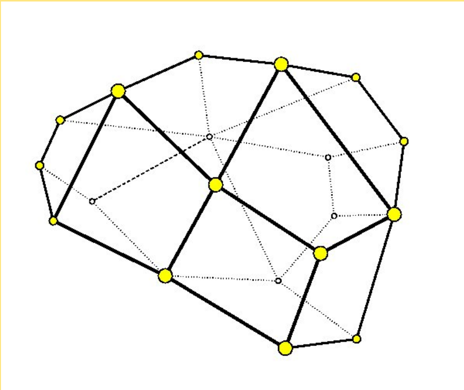


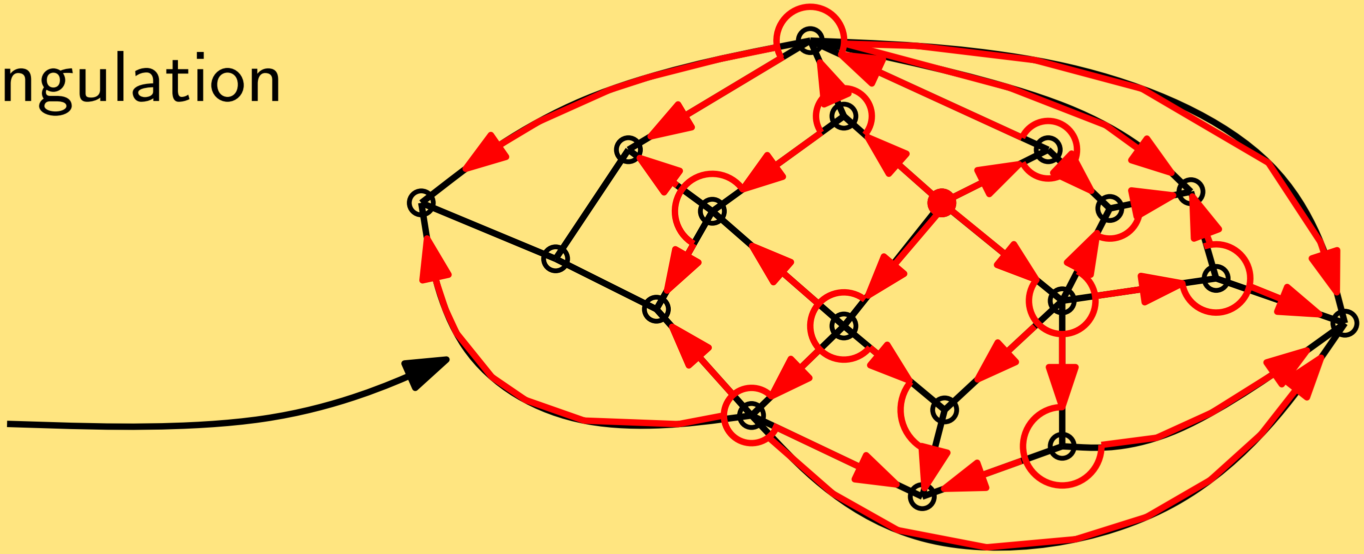
Apply bfs with the rotatoria rule
and cut along the flow



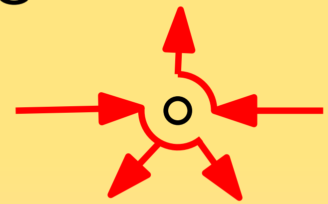
# Quadrangulations via breadth first search

Consider a planar quadrangulation



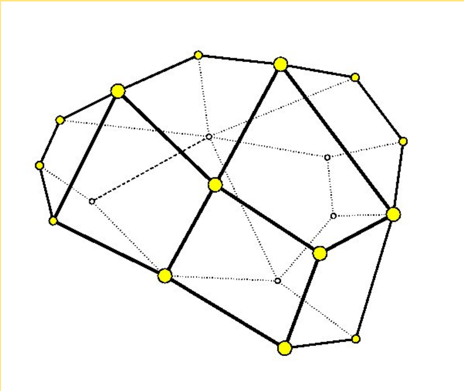


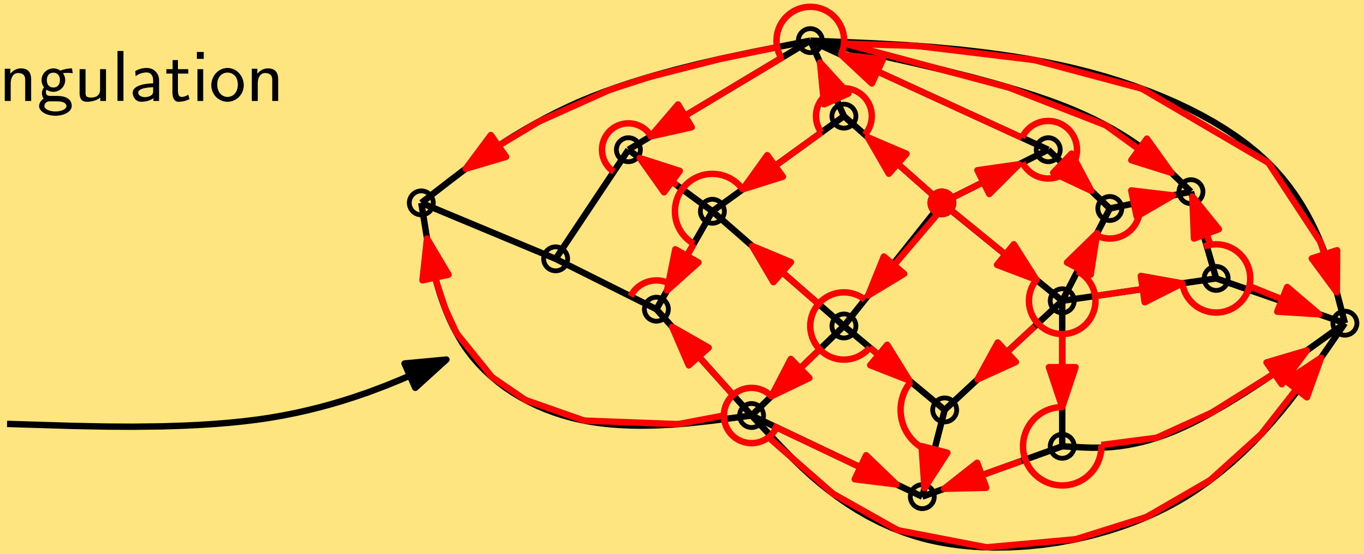
Apply bfs with the rotatoria rule
and cut along the flow



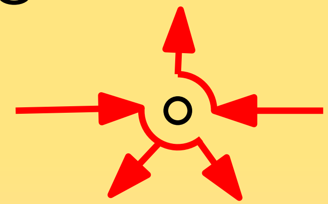
# Quadrangulations via breadth first search

Consider a planar quadrangulation



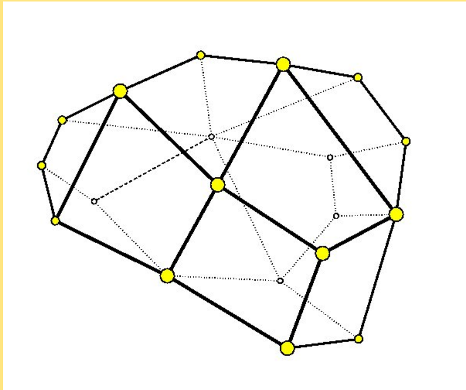


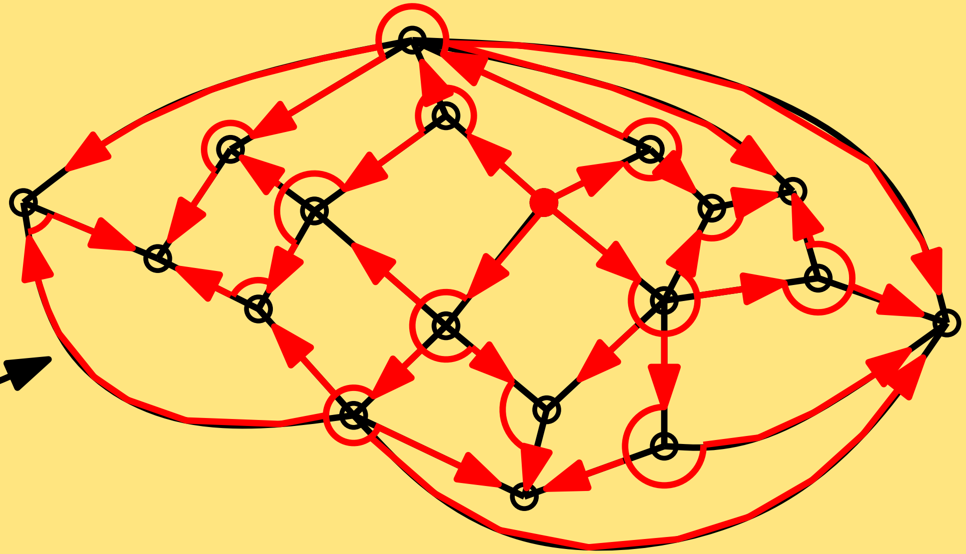
Apply bfs with the rotatoria rule and cut along the flow



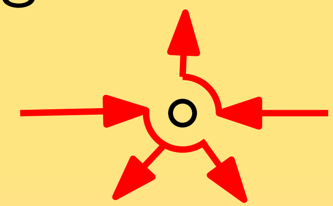
# Quadrangulations via breadth first search

Consider a planar quadrangulation



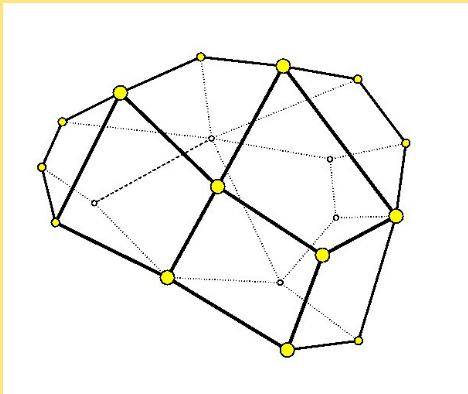


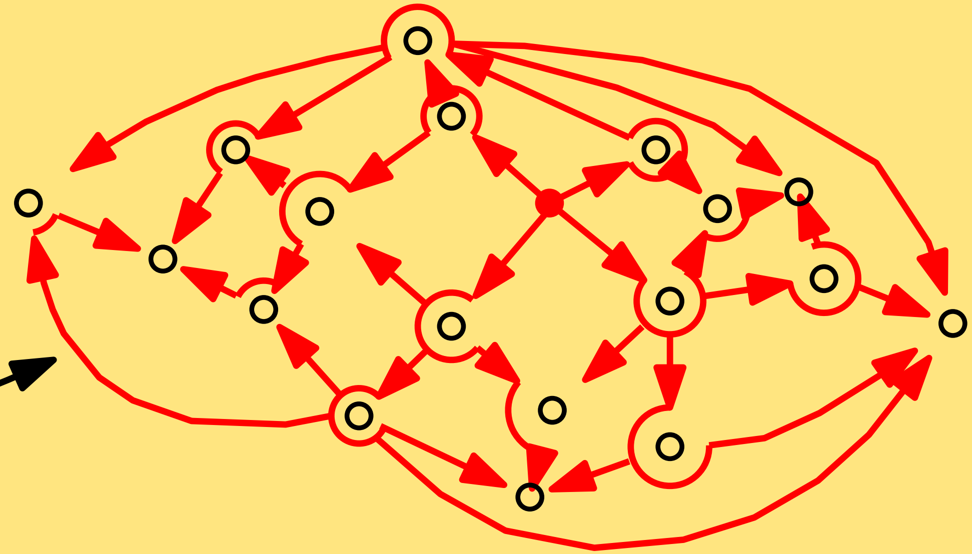
Apply bfs with the rotatoria rule
and cut along the flow



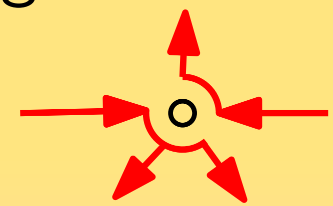
# Quadrangulations via breadth first search

Consider a planar quadrangulation



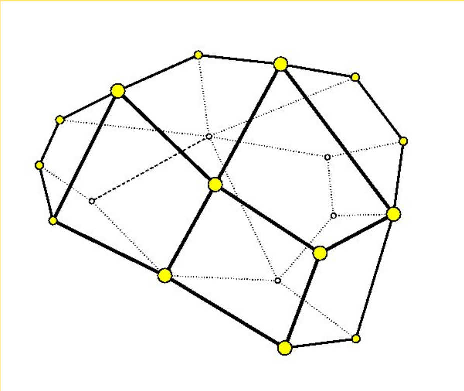


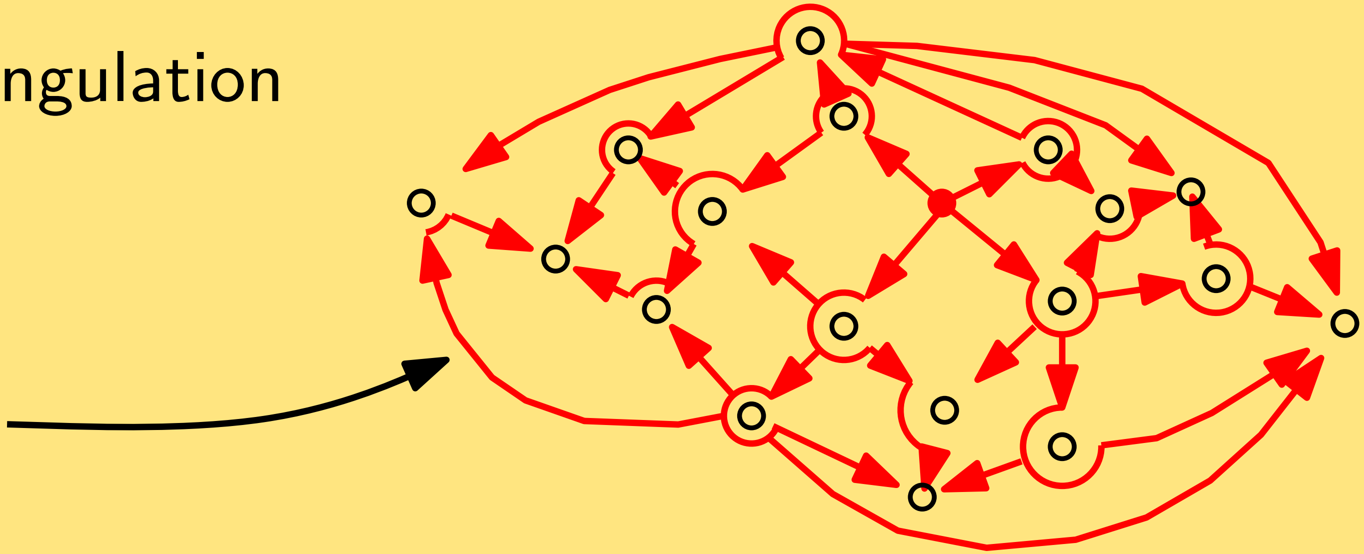
Apply bfs with the rotatoria rule
and cut along the flow



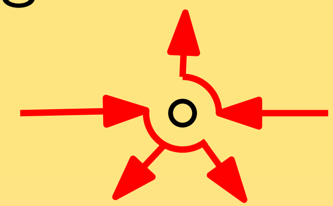
# Quadrangulations via breadth first search

Consider a planar quadrangulation

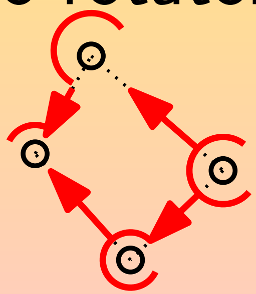




Apply bfs with the rotatoria rule
and cut along the flow

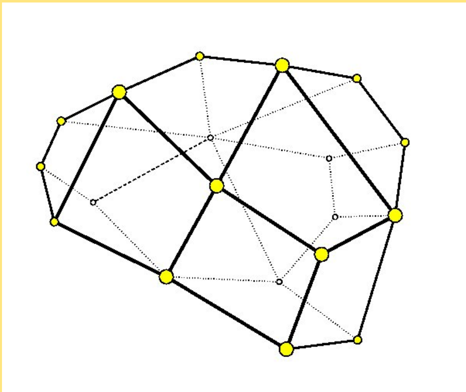


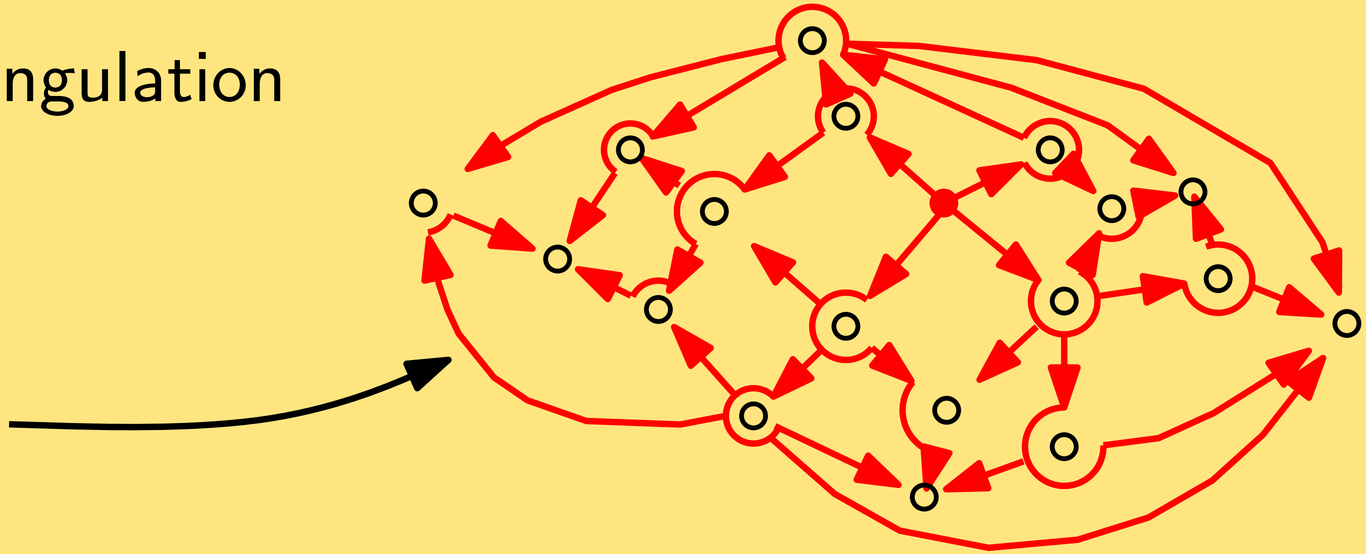
Each face sees exactly two rotatoria



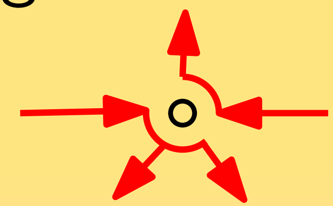
# Quadrangulations via breadth first search

Consider a planar quadrangulation

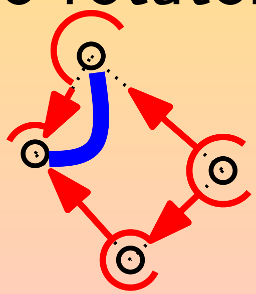




Apply bfs with the rotatoria rule
and cut along the flow



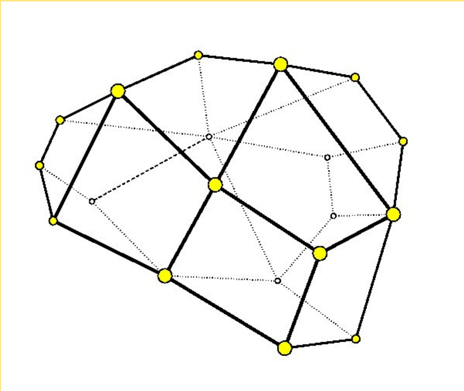
Each face sees exactly two rotatoria

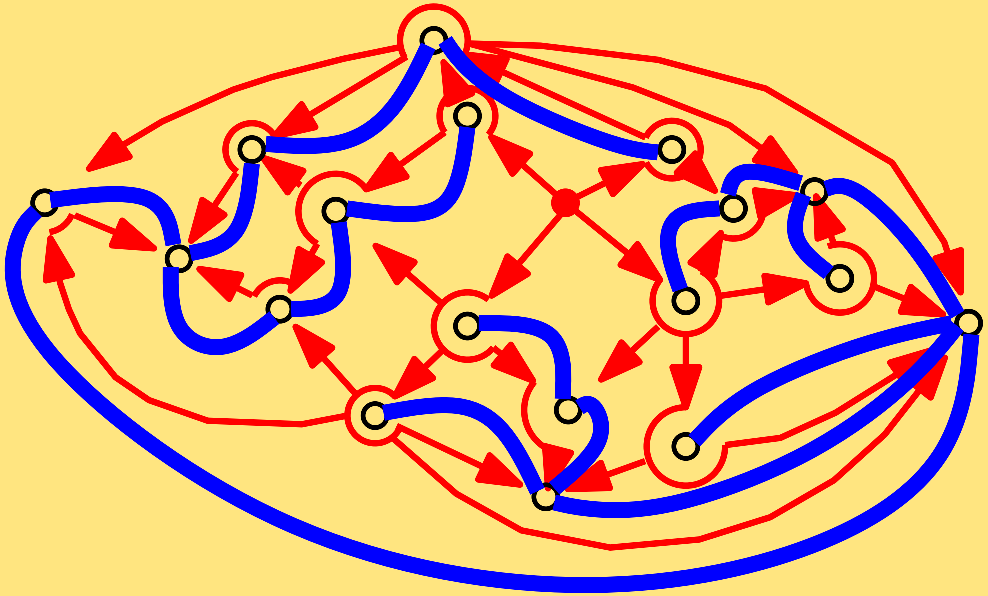


Join these 2 rotatoria!

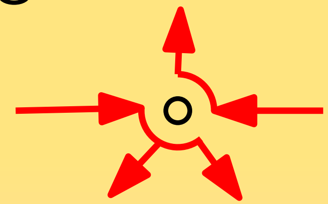
## Quadrangulations via breadth first search

Consider a planar quadrangulation



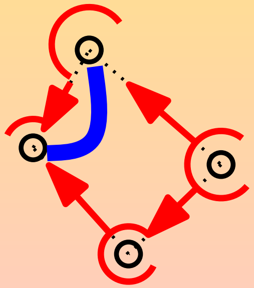


Apply bfs with the rotatoria rule and cut along the flow



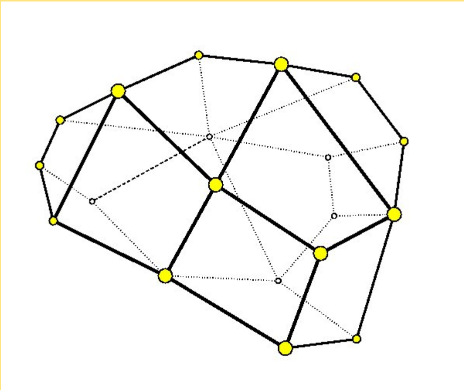
Each face sees exactly two rotatoria

Join these 2 rotatoria!

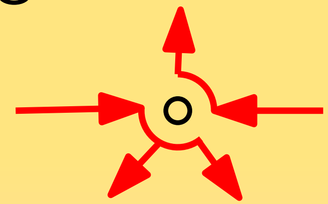


# Quadrangulations via breadth first search

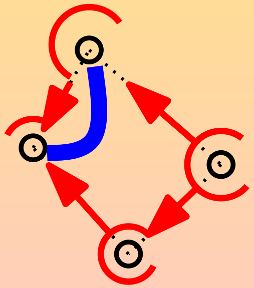
Consider a planar quadrangulation



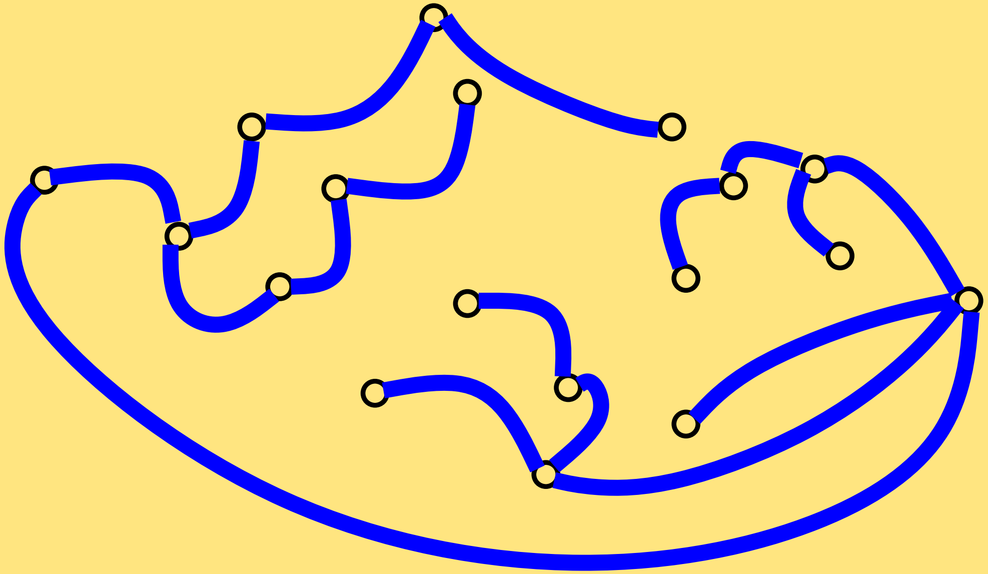
Apply bfs with the rotatoria rule and cut along the flow



Each face sees exactly two rotatoria



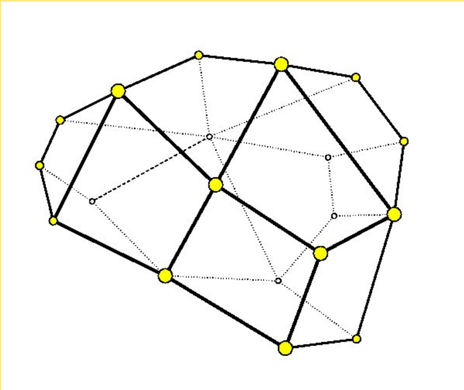
Join these 2 rotatoria!



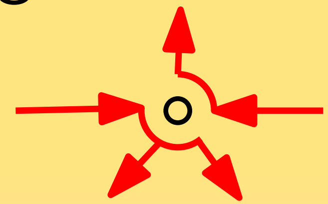
The result is **tree**.

# Quadrangulations via breadth first search

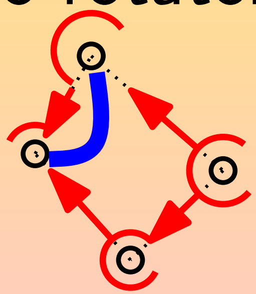
Consider a planar quadrangulation



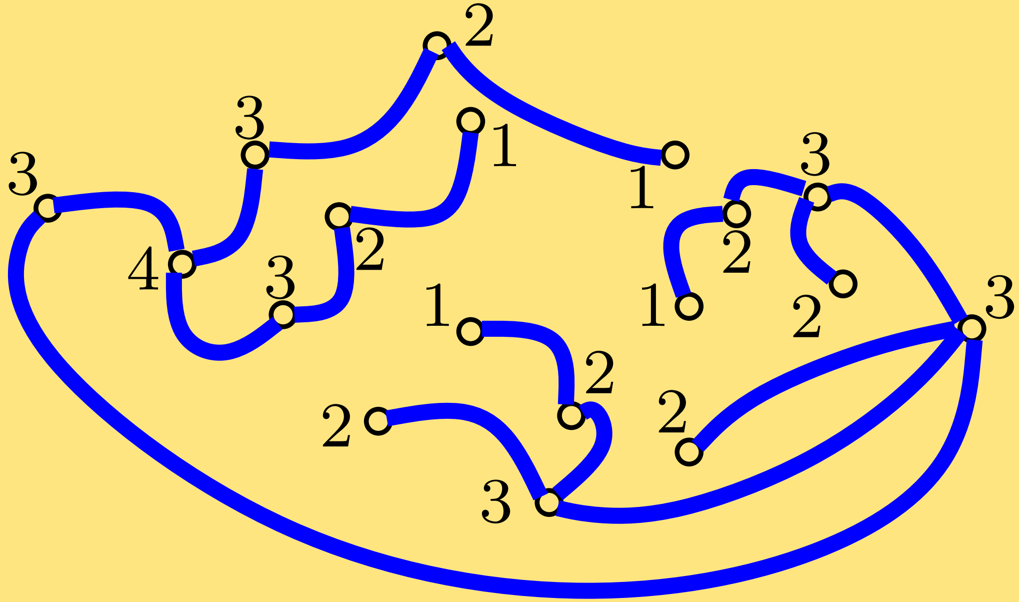
Apply bfs with the rotatoria rule and cut along the flow



Each face sees exactly two rotatoria



Join these 2 rotatoria!

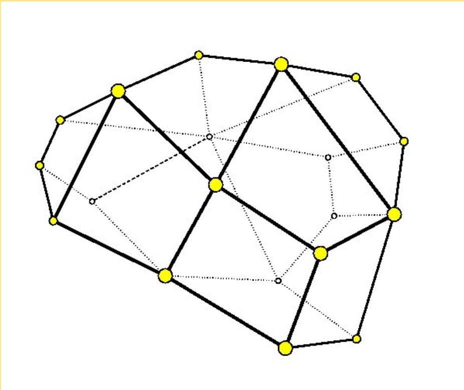


The result is **tree**.

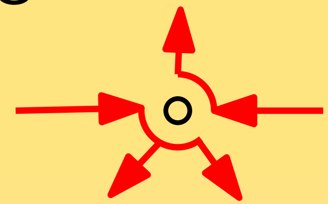
Label vertices by the round at which they were visited by bfs.

# Quadrangulations via breadth first search

Consider a planar quadrangulation

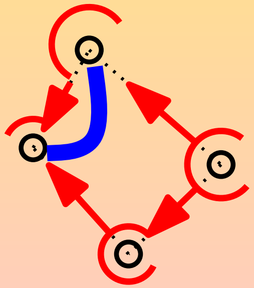


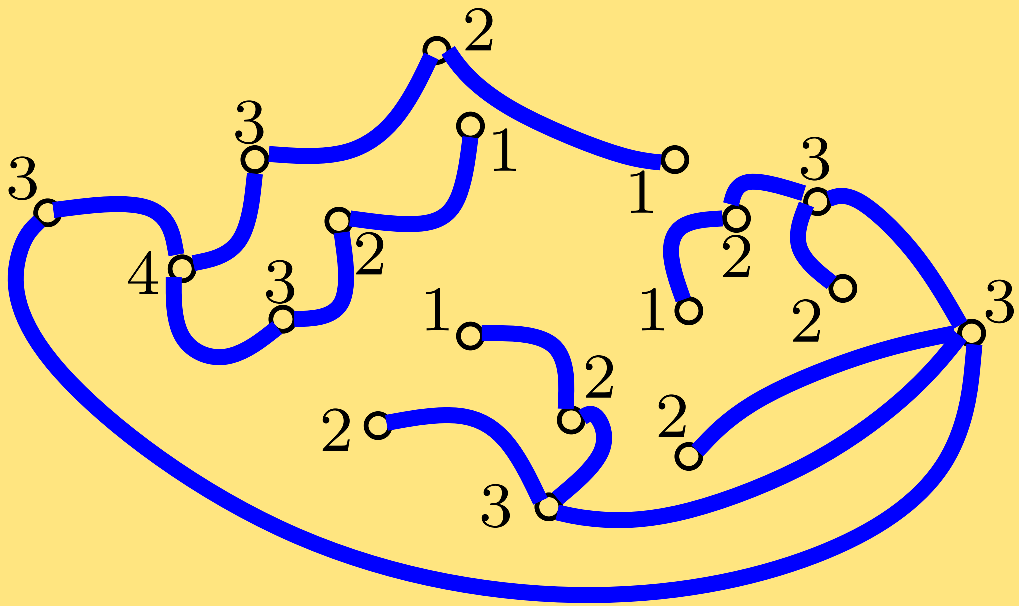
Apply bfs with the rotatoria rule and cut along the flow



Each face sees exactly two rotatoria

Join these 2 rotatoria!



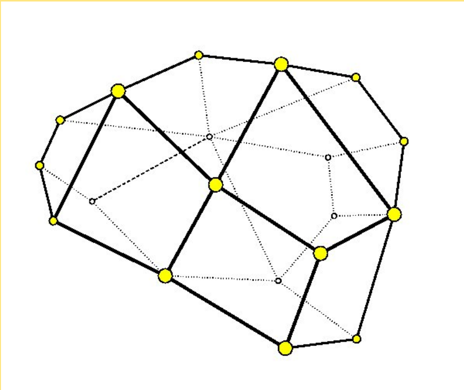



The result is a well labeled **tree**.

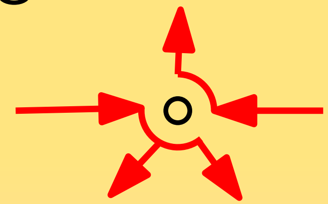
Label vertices by the round at which they were visited by bfs.

# Quadrangulations via breadth first search

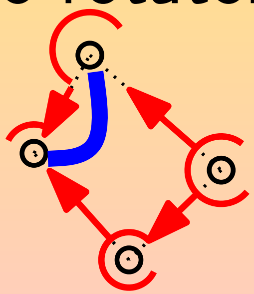
Consider a planar quadrangulation



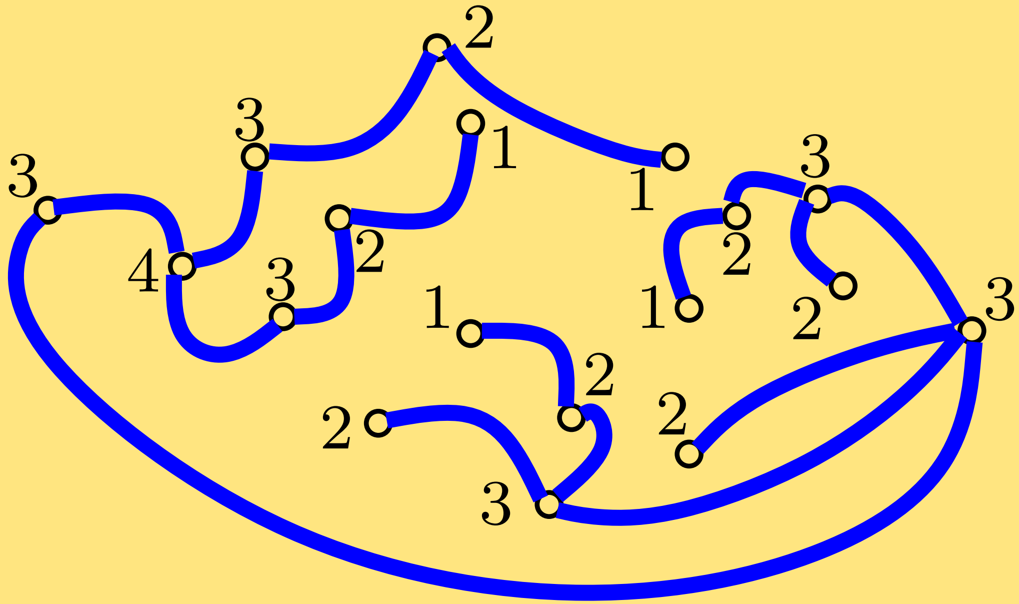
Apply bfs with the rotatoria rule and cut along the flow



Each face sees exactly two rotatoria



Join these 2 rotatoria!



The result is a well labeled **tree**.

Label vertices by the round at which they were visited by bfs.

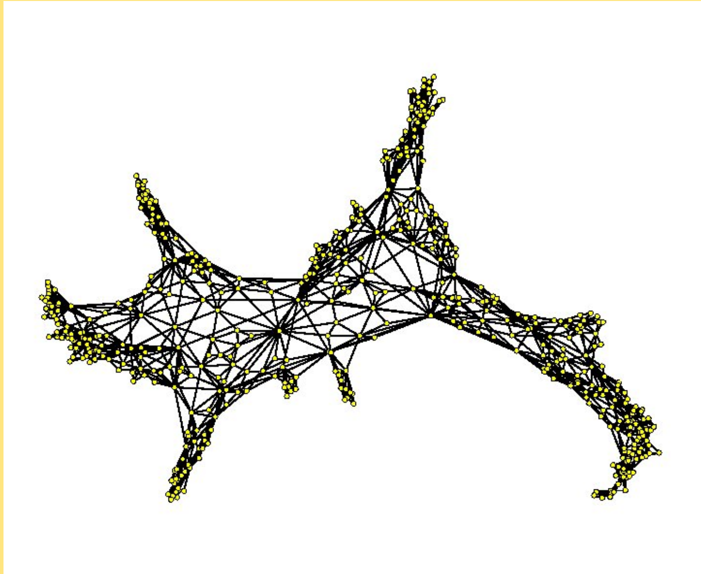
**Theorem**. This is a bijection.

$X_n$: pointed quads, $n$ faces

$\wr$

$T_n$: well labeled trees, $n$ vtx

# Quadrangulations via breadth first search

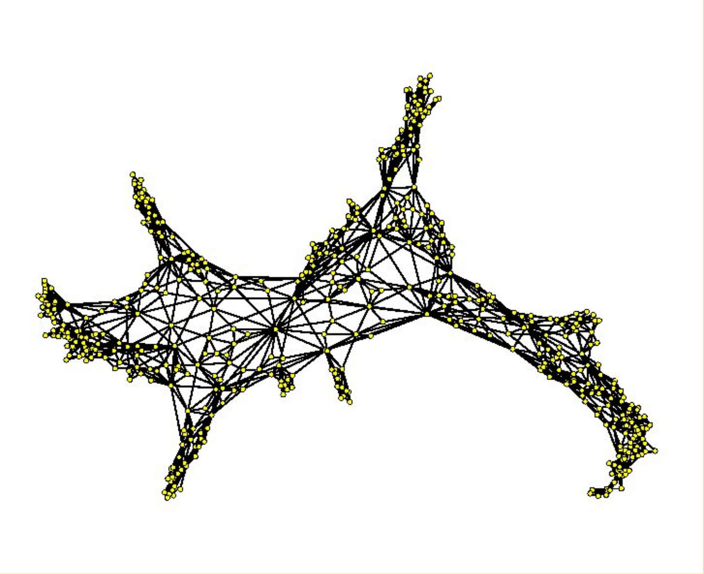


## use breadth first search to study the geometry

distance between 2 pts = nb of edges on a path

# Quadrangulations via breadth first search

use breadth first search to study the geometry



distance between 2 pts $=$ nb of edges on a path

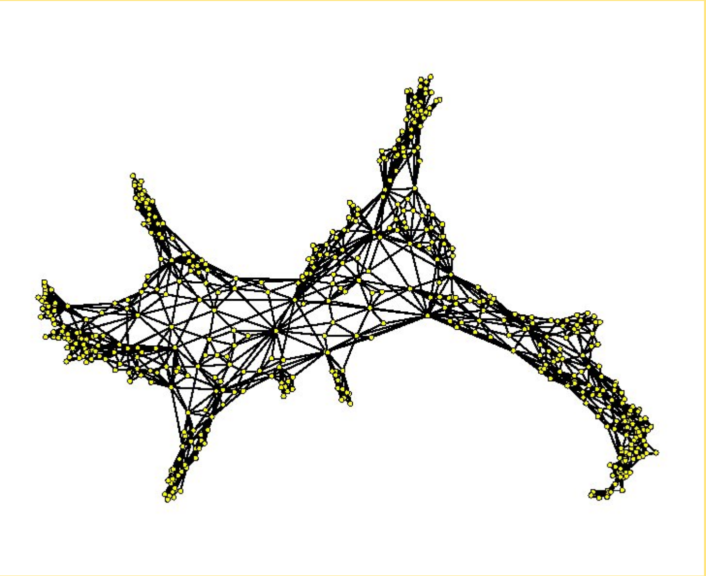
distance from basepoint

$=$ round of exploration by bfs

$\Rightarrow$ breadth first search computes distances:

# Quadrangulations via breadth first search

## use breadth first search to study the geometry



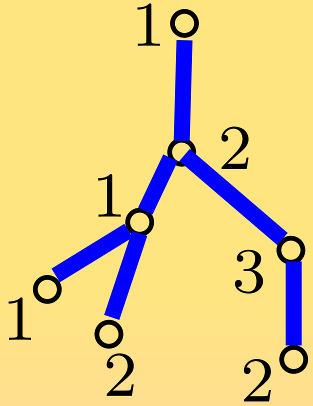
distance between 2 pts $=$ nb of edges on a path

distance from basepoint

$=$ round of exploration by bfs

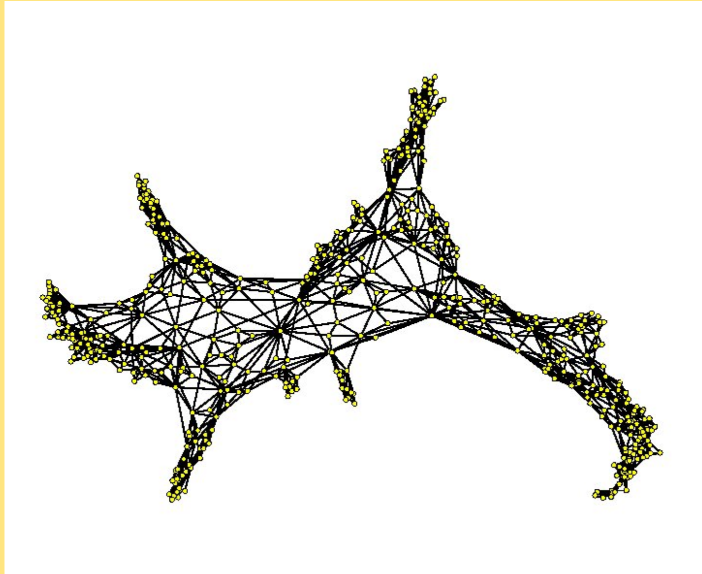
$\Rightarrow$ breadth first search computes distances:

- labels of the tree record distances from the basepoint



# Quadrangulations via breadth first search

## use breadth first search to study the geometry



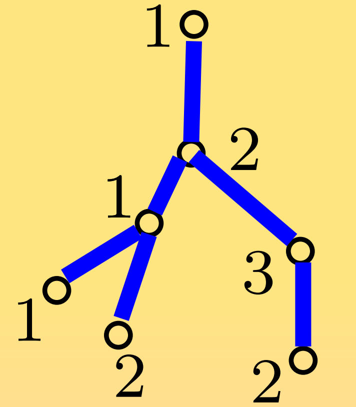
distance between 2 pts = nb of edges on a path

distance from basepoint

= round of exploration by bfs

$\Rightarrow$ breadth first search computes distances:

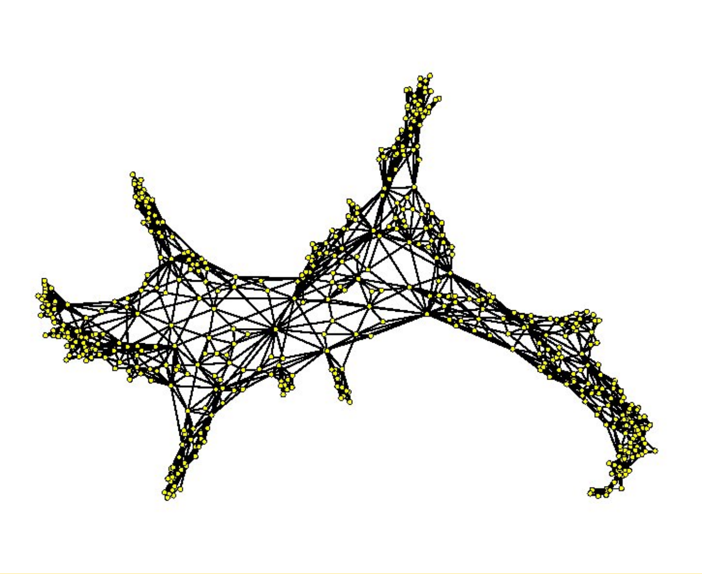
- labels of the tree record distances from the basepoint
- the height of a random tree of size $n$ is $n^{1/2}$
- the random walk of labels on a branch of length $\ell$ has max about $\ell^{1/2}$




$\Rightarrow$ typical labels are of order $n^{1/4}$.

# Quadrangulations via breadth first search

## use breadth first search to study the geometry



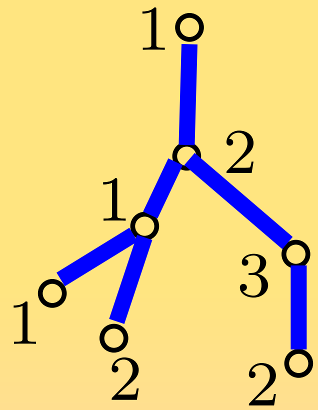
distance between 2 pts $=$ nb of edges on a path

distance from basepoint

$=$ round of exploration by bfs

$\Rightarrow$ breadth first search computes distances:

- labels of the tree record distances from the basepoint
- the height of a random tree of size $n$ is $n^{1/2}$
- the random walk of labels on a branch of length $\ell$ has max about $\ell^{1/2}$




$\Rightarrow$ typical labels are of order $n^{1/4}$.

**Theorem** (Chassaing-S, 2004).
The distance between 2 random vertices of $X_n$ is of order $n^{1/4}$.

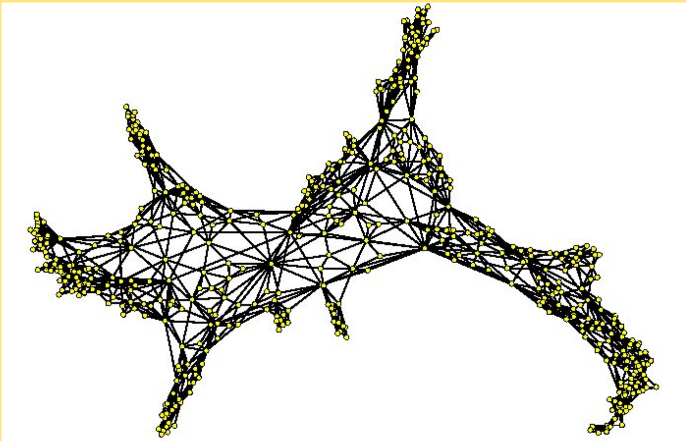
## Some properties of random discrete surfaces

This approach was pursued by Chassaing-Durhuus (2005), Marckert-Mokkadem (2004), Miermond (2005), Weill (2006)... culminating with

# Some properties of random discrete surfaces

This approach was pursued by Chassaing-Durhuus (2005), Marckert-Mokkadem (2004), Miermond (2005), Weill (2006)... culminating with

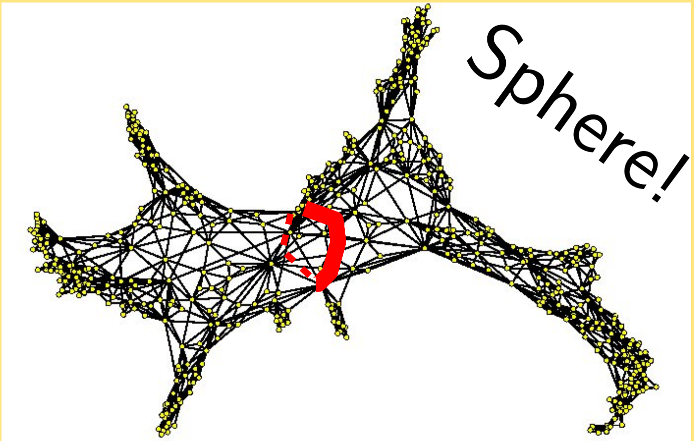
**Theorem** (Le Gall, 2006). Rescaled planar quadrangulations converge in the large size limit to a *random continuum planar map* that has spherical topology.



## Some properties of random discrete surfaces

This approach was pursued by Chassaing-Durhuus (2005), Marckert-Mokkadem (2004), Miermond (2005), Weill (2006)... culminating with

**Theorem** (Le Gall, 2006). Rescaled planar quadrangulations converge in the large size limit to a *random continuum planar map* that has spherical topology.

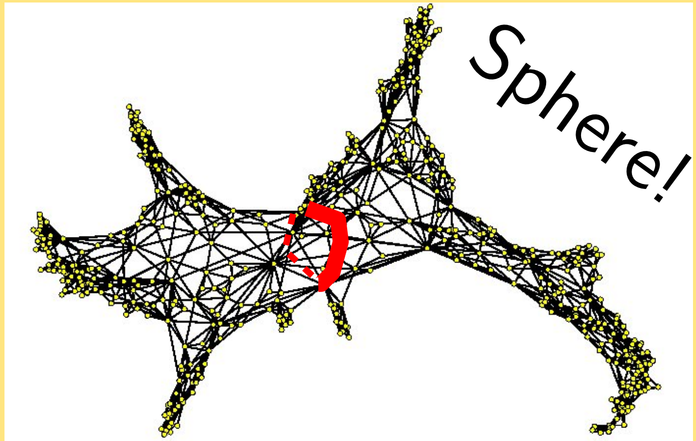



In particular there exists no separating cycle of size $\ll n^{1/4}$.

## Some properties of random discrete surfaces

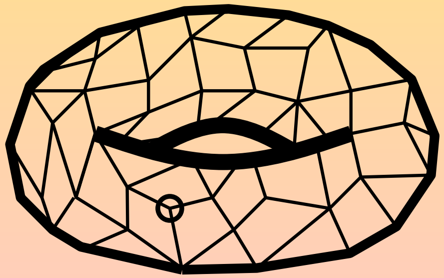
This approach was pursued by Chassaing-Durhuus (2005), Marckert-Mokkadem (2004), Miermond (2005), Weill (2006)... culminating with

**Theorem** (Le Gall, 2006). Rescaled planar quadrangulations converge in the large size limit to a *random continuum planar map* that has spherical topology.




In particular there exists no separating cycle of size $\ll n^{1/4}$.

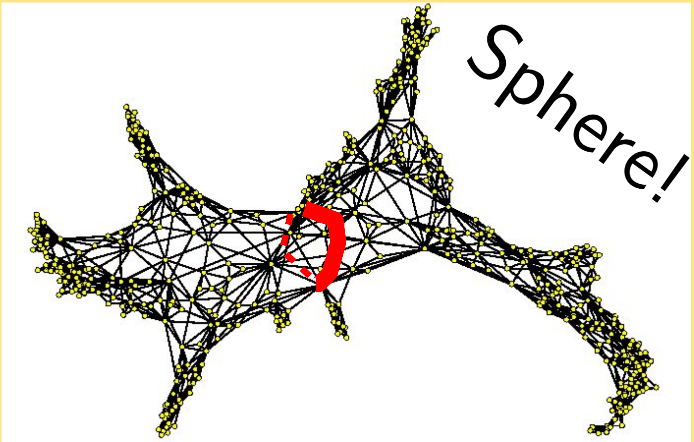
The bfs exploration works also for higer genus surfaces:
**Theorem** (Chapuy-Marcus-S. 2006) The distance between 2 random vertices of a random quad $X_n^g$ of genus $g$ is of order $n^{1/4}$.



## Some properties of random discrete surfaces

This approach was pursued by Chassaing-Durhuus (2005), Marckert-Mokkadem (2004), Miermond (2005), Weill (2006)... culminating with

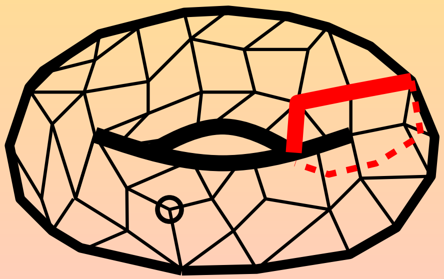
**Theorem** (Le Gall, 2006). Rescaled planar quadrangulations converge in the large size limit to a *random continuum planar map* that has spherical topology.




In particular there exists no separating cycle of size $\ll n^{1/4}$.

The bfs exploration works also for higer genus surfaces:
**Theorem** (Chapuy-Marcus-S. 2006) The distance between 2 random vertices of a random quad $X_n^g$ of genus $g$ is of order $n^{1/4}$.

**Conjectures.**
There is no non-contractible cycles with size $\ll n^{1/4}$.
The rescaled continuum limit exists and has genus $g$.



## A conjecture on random graphs with low genus

Let $Y_n^g$ be a uniform random connected labelled graphs with $n$ vertices that can be embedded on a surface of genus $g$.

For instance $Y_n^0$ is a random connected planar graph with $n$ vertices.

## A conjecture on random graphs with low genus

Let $Y_n^g$ be a uniform random connected labelled graphs with $n$ vertices that can be embedded on a surface of genus $g$.

For instance $Y_n^0$ is a random connected planar graph with $n$ vertices.

**Conjecture.** The graph $Y_n^g$ is a.s. composed of a 3-connected graph $Core(Y)$ of size $\Theta(n)$ with edges replaced by small planar networks and with small pending planar components.

Moreover $Core(Y)$ a.s. has minimal genus $g$ and has a unique minimal embedding. The small parts have size $O(n^{2/3})$.

In the rescaled limit, $Y_n^g$ converge to the same continuum random map of genus $g$ as $X_n^g$.

Cf. McDiarmid, Noy, Steger's talks for proofs...

Many thanks for your attention !

Many thanks to my collaborators!