LET'S COUNT THE CONNECTED CHORD DIAGRAMS

Séminaire Algo (LIGM)

Coauthors: Karen YEATS (Waterloo), Noam ZEILBERGER (Birmingham)
Part I

Interesting things to know about chord diagrams
**DEFINITIONS**

- **Diagram with** $n$ **chords**
  
  $= \text{matching of the set } \{1, \ldots, 2n\}$

- **Number of diagrams with** $n$ **chords**

  $= \text{? ? ?}$
DEFINITIONS

diagram with \( n \) chords

= matching of

the set \( \{1, \ldots, 2n\} \)

number of diagrams with \( n \) chords

\[ = (2n-1)!! \]

\[ = (2n-1) \times (2n-3) \times \ldots \times 3 \times 1 \]
DEFINITIONS

Diagram with $n$ chords = matching of the set $\{1, \ldots, 2n\}$

(more refined) enumeration of chord diagrams

computer science
theoretical physics
cumulants

knot theory
bioinformatics

...
DEFINITIONS

A diagram with $n$ chords is a matching of the set $\{1, \ldots, 2n\}$.

- Computer science
- Theoretical physics
- Cumulants
- Knot theory
- Bioinformatics
- (More refined) enumeration of chord diagrams...
Why counting diagrams...

In Computer Science?

Application 1

Queue priority (heap)

Operations:
- Add an element with value $\mathbf{a}$
- Remove the smallest element

History: $+8 +3 +2 +10 \quad \text{min} +3 \quad \text{min} +6 +1$
WHY COUNTING DIAGRAMS ... IN COMPUTER SCIENCE?

Application 1

Queue priority (heap)

Operations:

1. Add an element with value $k$
2. Remove the smallest element

History: $8 + 3 + 2 + 10 = \min 8 + 3 = \min 6 + 4$
Why counting diagrams ... in computer science?

Application 1

Queue priority (heap)

Operations:

- Add an element with value \( \ell \)
- Remove the smallest element

History: \( +_8 +_3 +_2 +_{10} =_{\text{min}} +_9 =_{\text{min}} +_6 +_4 \)
WHY COUNTING DIAGRAMS ... IN COMPUTER SCIENCE?

Application

Queue priority (heap)

Operations:

- Add an element with value \( k \)
- Remove the smallest element

History: \( 8 + 3 + 2 + 10 = \min 3 = \min 6 + 4 \)
Why counting diagrams in Computer Science?

Application 1: Queue priority (heap)

Operations:
- Add an element with value \( k \)
- Remove the smallest element

History: +8 +3 +2 +10 \( \text{min} \) +9 \( \text{min} \) +6 +4
WHY COUNTING DIAGRAMS ... IN COMPUTER SCIENCE?

Application

Queue priority (heap)

Operations:

+ \_ \_ Add an element with value \_ \_ 

- _ _ Remove the smallest element

History: +8 +3 +2 +10 = min +3 = min +6 +4
WHY COUNTING DIAGRAMS ... IN COMPUTER SCIENCE?

Application 1

Queue priority (heap)

Operations:

- Add an element with value $a$
- Remove the smallest element

History: $+8 +3 +2 +10 = \text{min} +9 = \text{min} +6 +4$
Why counting diagrams in computer science?

Application

Queue priority (heap)

Operations:
- Add an element with value \( k \)
- Remove the smallest element

History: +8 +3 +2 +10 = min +9 = min +6 +4
Why counting diagrams... in computer science?

Application

Queue priority (heap)

Operations:

- Add an element with value $k$
- Remove the smallest element

History: $+8 +3 +2 +10 - \min -9 - \min -6 -1$
Why counting diagrams ... in Computer Science?

Application 1: Queue priority (heap)

Operations:
- Add an element with value \( k \)
- Remove the smallest element

History: \( +_2 +_3 +_8 +_{10} = \text{min} +_3 = \text{min} +_6 +_1 \)
WHY COUNTING DIAGRAMS ... IN COMPUTER SCIENCE?

Application 1

Queue priority (heap)

Operations:

- Add an element with position $k$, $1 \leq k \leq \text{size queue} + 1$
- Remove the smallest element

History: $+1 +1 +1 +4 \rightarrow \text{min} +3 \rightarrow \text{min} +1 +1$

Now we focus on relative positions
Why counting diagrams ... in Computer Science?

Application 1

Queue priority (heap)

Operations:

+ Add an element \( k \) with position \( k \)
  \( 1 \leq k \leq \text{size queue} + 1 \)

- Remove the smallest element

History: \( +_1 +_1 +_1 +_4 \) \( \text{min}_3 \) \( \text{min}_1 +_1 +_1 \)

Now we focus on relative positions
Why counting diagrams in computer science?

Application

Queue priority (heap)

Operations:

- Add an element $k$ with position $k$, $1 \leq k \leq \text{size queue} + 1$
- Remove the smallest element

History: $\mathbf{+_1+_1+_1+_4 = \text{min}+_3 = \text{min}+_1+_1}$

Now we focus on relative positions.
WHY COUNTING DIAGRAMS ... IN COMPUTER SCIENCE?

Application

Queue priority (heap)

Operations:
- Add an element \( k \) with position \( k \), \( 1 \leq k \leq \text{size queue} + 1 \)
- Remove the smallest element

History: \( +1 +1 +1 +4 \) - \( \min +3 \) - \( \min +1 +1 \)

Now we focus on relative positions
Why counting diagrams ...
In computer science?

Queue priority (heap)

Operations:

- Add an element \( k \) with position \( k \)
  \( 1 \leq k \leq \text{size queue} + 1 \)
- Remove the smallest element

Application

History:

Now we focus on relative positions
Why counting diagrams ... in computer science?

Application

Queue priority (heap)

Operations:
- Add an element \( k \) with position \( k \), \( 1 \leq k \leq \text{size queue} + 1 \)
- Remove the smallest element

History: \( +_1 +_1 +_1 +_4 \rightarrow ^{\min} +_3 \rightarrow ^{\min} +_1 +_1 \)

Now we focus on relative positions
Why counting diagrams... in computer science?

Queue priority (heap)

Operations:

- Add an element $k$ with position $k$
  
  $1 \leq k \leq \text{size queue} + 1$

- Remove the smallest element

History: $\oplus_1 \oplus_1 \oplus_1 \oplus_4 \oplus_{\min} \oplus_3 \oplus_{\min} \oplus_1 \oplus_1$

Now we focus on relative positions
WHY COUNTING DIAGRAMS ... IN COMPUTER SCIENCE?

Application

Queue priority (heap)

Operations:

+ Add an element $k$ with position $k$
  \[ 1 \leq k \leq \text{size queue} + 1 \]

- Remove the smallest element

History: \[ + + + + + + _{\text{min}} + + _{\text{min}} + + \]

Now we focus on relative positions
WHY COUNTING DIAGRAMS ... IN COMPUTER SCIENCE?

Application

Queue priority (heap)

Operations:

+ Add an element \( k \) with position \( k \)
  \[ 1 \leq k \leq \text{size queue} + 1 \]

- Remove the smallest element

History: + + + + + + + -\_min + + + + + Now we focus on relative positions
Why counting diagrams ... in computer science?

Application

Queue priority (heap)

Operations:

- Add an element $\mathbf{k}$ with position $\mathbf{k}$, $1 \leq \mathbf{k} \leq \text{size queue} + 1$

- Remove the smallest element

History:

Now we focus on relative positions
WHY COUNTING DIAGRAMS ... 
IN COMPUTER SCIENCE?

Application
Queue priority (heap)

[Flajolet Françon Vuillemin] Histories ending at 0 ↔ Chord diagrams

Diagram:

History: $+_{1} +_{1} +_{1} +_{4}$ $\overset{\text{min}}{+_{3}} +_{1} +_{1} +_{1} +_{1} +_{1} +_{1} +_{1}$
Why counting diagrams ... in computer science?

Application 2

Biased model $G(n, m)$

Random diagram $\rightarrow$ Intersection graph

$n = \text{number of chords}$

$m = \text{number of crossings}$

$n = \text{number of vertices}$

$m = \text{number of edges}$
DEFINITIONS

diagram with $n$ chords
= matching of
the set $\{1, \ldots, 2n\}$

connected diagram =
“everything holds in one block.”
DEFINITIONS

diagram with $n$ chords = matching of the set \( \{1, \ldots, 2n\} \)

connected diagram = “everything holds in one block.”

3 connected components
**DEFINITIONS**

Diagram with $n$ chords: matching of the set $\{1, \ldots, 2n\}$

Connected diagram = “everything holds in one block.”

3 connected components:

1. $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6$
2. $7 \rightarrow 8 \rightarrow 9$
3. $10$
DEFINITIONS

diagram with \( n \) chords = matching of the set \( \{1, \ldots, 2n\} \)

connected diagram = "everything holds in one block."
DEFINITIONS

Enumeration of connected diagrams:

PREHISTORY
- 1952 Touchard

EXPLICIT FORMULAS
- 1978 Stein Everett

ANALYTIC COMBINATORICS
- 2000 Flajolet Noy

connected diagram = “everything holds in one block.”
Elementary Enumeration

Number of connected diagrams with $n$ chord = $C_n$

$C_1 = 1$, $C_2 = 1$, $C_3 = 4$, $C_4 = 27$, $C_5 = 248$

For $n = 3$:

Induction formula [Stein] $C_n = (n-1) \sum_{k=1}^{n-1} C_k \times C_{n-k}$
Proof of $c_n = (n-1) \times \sum_{k=1}^{n-1} c_k \times c_{n-k}$.
Proof of \[ c_n = \sum_{k=1}^{n-1} (2k-1) \times c_k \times c_{n-k} \]

\[ \text{Formula: } c_n = (n-1) \times \sum_{k=1}^{n-1} c_k \times c_{n-k} \]
Proof of \[ c_n = \sum_{k=1}^{n-1} (2k-1) \cdot c_k \cdot c_{n-k} \]

\[ c_n = \sum_{k=1}^{n-1} (2(n-k)-1) \cdot c_{n-k} \cdot c_k \]

Formula: \[ c_n = (n-1) \times \sum_{k=1}^{n-1} c_k \cdot c_{n-k} \]
Elementary Enumeration

Proof of \( \mathcal{C}_n = \sum_{k=1}^{n-1} (2k-1) \times c_k \times c_{n-k} \quad ? \)

\[ + \]

\[ \mathcal{C}_n = \sum_{k=1}^{n-1} (2(n-k)-1) \times c_{n-k} \times c_k \]

\[ \underline{\sum_{k=1}^{n-1} (2n-2) \times c_k \times c_{n-k}} \]

\[ 2 \mathcal{C}_n = \sum_{k=1}^{n-1} (2n-2) \times c_k \times c_{n-k} \]

Formula: \( \mathcal{C}_n = (n-1) \times \sum_{k=1}^{n-1} c_k \times c_{n-k} \)
Proof of \( \mathcal{C}_n = \sum_{k=1}^{n-1} (2k-1) \times c_k \times \mathcal{C}_{n-k} \) ?

\[ \mathcal{C}_n = \sum_{k=1}^{n-1} (2(n-k)-1) \times c_{n-k} \times c_k \]

\[ 2 \mathcal{C}_n = \sum_{k=1}^{n-1} (2n-2) \times c_k \times \mathcal{C}_{n-k} \]

\( \% 2 \)

Formula: \( \mathcal{C}_n = (n-1) \times \sum_{k=1}^{n-1} c_k \times \mathcal{C}_{n-k} \)
Elementary Enumeration

Proof of $c_n = \sum_{k=1}^{n-1} (2k-1) \times c_k \times c_{n-k}$?
Proof of $c_n = \sum_{k=1}^{n-1} (2k-1) \times c_k \times c_{n-k}$ ?
Proof of \( c_n = \sum_{k=1}^{n-1} (2k-1) \times c_k \times c_{n-k} \) ?
Proof of \[ c_n = \sum_{k=1}^{n-1} (2k-1) \times c_k \times c_{n-k} \]?
Proof of \( c_n = \sum_{k=1}^{n-1} (2k-1) \times c_k \times c_{n-k} \)?
ELE\text{MENTARY ENUMERATION}

Proof of \[ c_n = \sum_{k=1}^{n-1} (2k-1) \times c_k \times c_{n-k} \]
ELEMNETARY ENUMERATION

Proof of $c_n = \sum_{k=1}^{n-1} (2k-1)c_k c_{n-k}$?
Elementary Enumeration

Proof of \( c_n = \sum_{k=1}^{n-1} (2k-1) \times c_k \times c_{n-k} \)?
Proof of $c_n = \sum_{k=1}^{n-1} (2k-1) \times c_k \times c_{n-k}$.
ASYMPTOTIC BEHAVIOR

\[ c_n \sim \frac{1}{e} \times (2n-1)!! \]

[Stein Everett]

Corollary: \[ P(\text{connected diagram}) = \frac{c_n}{(2n-1)!!} \rightarrow \frac{1}{e} \]
ASYMPTOTIC BEHAVIOR

\[ c_n \sim \frac{1}{e} \times (2n-1)!! \]

[Stein Everett]

Corollary: \[ P(\text{connected diagram}) = \frac{c_n}{(2n-1)!!} \rightarrow \frac{1}{e} \]

Better \[ \text{[Flajolet Noy]} \]

- number of connected components \( \sim \text{Poisson}(1) \)
- \( |C| \) - size of the largest component \( \sim \text{Poisson}(1) \)
Part II

Analytic side: terminal chords

with Karen Yeats
DEFINITION

terminal chord =
chord \((a, b)\) such that
if \((c, d)\) intersects \((a, b)\),
then \(c < a\).
**Definition**

terminal chord = chord \((a,b)\) such that if \((c,d)\) intersects \((a,b)\), then \(c < a\).

**Question:** Average number of terminal chords in a connected chord diagram?
The Dyson–Schwinger equation

\[ G(x, L) = 1 - x \, G(x, \frac{\partial}{\partial x})^{-1} \left( e^{-Lp} - 1 \right) F(p) \bigg|_{p=0} \]

has for solution:

\[ G(x, L) = 1 - \sum_{i \geq 1} \sum_{C \text{ connected chord diagram}} \frac{L^i}{i!} \cdot x^{1 \cdot c_1 - k} \cdot f_0 \cdot f_{t_1 - i} \cdot f_{t_2 - t_1} \cdot f_{t_3 - t_2} \cdots f_{t_k - t_{k-1}} \]

where \( t_1 < t_2 < \ldots < t_k \) denote the positions of terminal chords of \( C \) for the intersection order.
The Dyson-Schwinger equation

\[ G(x,L) = 1 - x \left[ G(x, \frac{\partial}{\partial x})^{-1} \left( e^{-L} - 1 \right) \right]_{x=0} \]

has for solution:

\[ G(x,L) = 1 - \sum_{i \geq 1} \sum_{C \text{ connected}} \frac{L^i}{i!} \alpha^i \beta_0^{1\text{cl}} \beta_1 \cdots \beta_{k-i} \cdots \beta_n \]

where \( \beta_1, \beta_2, \ldots, \beta_k \) denote the positions of terminal chords of \( C \) for the intersection order.
WHY IT IS HARD

The generating function

\[ C(z, u) = \sum_{n \geq 0} \sum_{k \geq 0} c_{n,k} z^n u^k \]

where \( c_{n,k} \) = number of connected diagrams with \( n \) chords and \( k \) terminal chords satisfies

\[ C(z, u) = z u + \frac{z}{8} \left( 2 z \frac{d^2 C(z, u)}{dz^2} - C(z, u) \right) \frac{1 - 2 z \frac{d C(z, u)}{dz} + C(z, u)}{1 - 2 z \frac{d C(z, u)}{dz} + C(z, u)} . \]
WHY IT IS HARD

The generating function \( C(z, \mu) = \sum_{n \geq 0} \sum_{k \geq 0} c_{n, k} z^n \mu^k \) where \( c_{n, k} \) = number of connected diagrams with \( n \) chords and \( k \) terminal chords satisfies

\[
C(z, \mu) = z \mu + \frac{z \left( 2z \frac{d}{dz} C(z, \mu) - C(z, \mu) \right)}{1 - 2z \frac{d}{dz} C(z, \mu) + C(z, \mu)}.
\]

non-linear differential equation
WHY IT IS HARD

The generating function $C(z, u) = \sum_{n \geq 0} \sum_{k \geq 0} c_{n,k} z^n u^k$ where $c_{n,k}$ is the number of connected diagrams with $n$ chords and $k$ terminal chords satisfies

$$C(z, u) = z u + \frac{z^2 (2 z \frac{\partial C}{\partial z} (z, u) - C(z, u))}{1 - 2 z \frac{\partial C}{\partial z} (z, u) + C(z, u)}.$$ 

Non-linear differential equation:

$$c_n = (n-1) \sum_{k=1}^{n-1} c_k c_{n-k} \Rightarrow c_n \geq n!.$$
AVERAGE NUMBER OF TERMINAL CHORDS

Idea:

large number of chords

proba = ??

or
AVERAGE NUMBER OF TERMINAL CHORDS

Idea:

\[
\text{large number of chords}
\]

\[
\text{proba} = \frac{(2n - 3) \cdot c_{n-1}}{c_n}
\]
Average number of terminal chords

**Idea:**

\[ \text{large number of chords} \]

\[ \text{proba} = \frac{(2n - 3) c_{n-1}}{c_n} \rightarrow 1 \]

Almost surely, removing the root chord does not disconnect the diagram.

Interesting but not enough! 😞
AVERAGE NUMBER OF TERMINAL CHORDS

Idea:

large number of chords

= 

or

or

or

or

something else
AVERAGE NUMBER OF TERMINAL CHORDS

Idea:

large number of chords

\[ \text{proba} = (2n-3) \frac{\xi_{n-1}}{\xi_n} = 1 - \frac{1}{n} + o\left(\frac{1}{n}\right) \]

or

\[ \text{proba} = (2n-5) \frac{\xi_{n-2}}{\xi_n} \sim \frac{1}{2n} \]

or

\[ \text{proba} = (2n-5) \frac{\xi_{n-2}}{\xi_n} \sim \frac{1}{2n} \]

or

something else

\[ \text{proba} = o\left(\frac{1}{n}\right) \]
Average number of terminal chords

Idea:

large number of chords

\[
\text{proba} = (2n-3) \frac{\zeta_{n-1}}{\zeta_n} = 1 - \frac{1}{n} + o\left(\frac{1}{n}\right)
\]

or

\[
\text{proba} = (2n-5) \frac{\zeta_{n-2}}{\zeta_n} \sim \frac{1}{2n}
\]

or

or

or

\[
\text{proba} = (2n-5) \frac{\zeta_{n-2}}{\zeta_n} \sim \frac{1}{2n}
\]

something else

\[
\text{proba} = o\left(\frac{1}{n}\right)
\]
AVERAGE NUMBER OF TERMINAL CHORDS

Idea:

large number of chords

\[ \text{proba} = (2n-3) \frac{\xi_{n-1}}{\xi_n} = 1 - \frac{1}{n} + O\left(\frac{1}{n}\right) \]

or

\[ \text{proba} = (2n-5) \frac{\xi_{n-2}}{\xi_n} \sim \frac{1}{2n} \]

or

something else

\[ \text{proba} = O\left(\frac{1}{n}\right) \]

It works! 😊
AVERAGE NUMBER OF TERMINAL CHORDS

Let $X_n$ be the random variable such that

$$X_n = \begin{cases} 
X_{n-1} & \text{with proba } 1 - \frac{1}{n} \\
X_{n-2} + 1 & \text{with proba } \frac{1}{n}
\end{cases}$$

Idea n°1

For every initial conditions,

$$X_n \rightarrow \text{Gaussian law.}$$

Idea n°2

Number of terminal chords 'n' $X_n$. 
Theorem

For the uniform distribution,

Number of terminal chords \[\overset{\text{distribution}}{\rightarrow}\] Gaussian law

mean \(\sim \ln(n)\)

variance \(\sim \ln(n)\)
 ORDRE D'INTERSECTION

Règle:

- petite corde
- corde racine
- puis celle composante
  - puis celle composante
  - enfin celle-là

Exemple:

1  3
2  6  7
4
5

1  2  3  4  5  6  7  8  9  10  11  12  13  14
NOMBRE MOYEN DE CORDES TERMINALES CONSECUTIVES

Si les cordes terminales sont en position $t_1 < t_2 < \ldots < t_k$, combien de $j$ satisfont $t_j - t_{j-1} = 1$ ?

ou

ou

autre chose

Théorème Pour la distribution uniforme, nombre de cordes terminales consécutives loi

Loi gaussienne

moyenne $\sim \frac{1}{2} \ln(n)$

variance $\sim \frac{1}{2} \ln(n)$
Current Work

→ Extension to other Dyson-Schwinger equations

  notion of decorated diagram
  with Yeats

→ Generalization of the method to other combinatorial families

  when the generating function is not analytic
  and satisfies a non-linear differential equation

  with Bodini & Dougal.
Part III

Bijective side: combinatorial maps

with Karen Yeats and Noam Zeilberger
AN OTHER FAMILY OF OBJECTS

combinatorial map = graph where we have cyclically ordered the 1/2-edges around each vertex.

Example. Four different maps:
ANOTHER FAMILY OF OBJECTS

Combinatorial map = graph where we have cyclically ordered the 1/2-edges around each vertex.

Example. Four different maps:

We root a map by marking a leaf.
ANOTHER FAMILY OF OBJECTS

combinatorial map = graph where we have cyclically ordered the 1/2-edges around each vertex.

1 edge

2 edges

3 edges
AN OTHER FAMILY OF OBJECTS

bridge = edge \neq 1 \text{ whose deletion disconnects the map.}

1 edge

2 edges

3 edges
ANOTHER FAMILY OF OBJECTS

\[ \text{bridge} = \text{edge} \neq 1 \text{ whose deletion disconnects the map.} \]

1 edge

2 edges

3 edges
Theorem

number of bridgeless maps with $n$ edges = number of connected diagrams with $n$ chords
Theorem

number of bridgeless maps with $n$ edges =

number of connected diagrams with $n$ chords

Reminder

$$c_n = \sum_{k=1}^{n-1} (2k-1)c_k c_{n-k}$$
Theorem

Number of bridgeless maps with $n$ edges $=$ number of connected diagrams with $n$ chords

Reminder

$$c_n = \sum_{k=1}^{n-1} (2k-1) c_k c_{n-k}$$
Theorem

number of bridgeless maps with n edges = number of connected diagrams with n chords

Reminder

\[ c_n = \sum_{k=1}^{n-1} (2k-1)c_k c_{n-k} \]
Theorem

number of bridgeless maps with $n$ edges = number of connected diagrams with $n$ chords

\[ \text{Reminder} \quad c_n = \sum_{k=1}^{n-1} (2k-1) c_k c_{n-k} \]
Theorem

number of bridgeless maps
with $n$ edges

= number of connected diagrams
with $n$ chords

Reminder $C_n = \sum_{k=1}^{n-1} (2k+1) C_k C_{n-k}$
Theorem

number of bridgeless maps with $n$ edges = number of connected diagrams with $n$ chords

Reminder

$$c_n = \sum_{k=1}^{n-1} (2k-1) c_k c_{n-k}$$
**Theorem**

number of bridgeless maps with $n$ edges = number of connected diagrams with $n$ chords

**Reminder**

$$c_n = \sum_{k=1}^{n-1} (2k-1)c_k c_{n-k}$$
**Theorem**

number of bridgeless maps with \( n \) edges = number of connected diagrams with \( n \) chords

**Reminder**

\[
c_n = \sum_{k=1}^{n-1} (2k-1) c_k c_{n-k}
\]
Theorem

number of bridgeless maps with \( n \) edges

= 

number of connected diagrams with \( n \) chords

\[
\text{Reminder} \quad c_n = \sum_{k=1}^{n-1} (2k-1)c_k c_{n-k}
\]
Theorem

number of bridgeless maps with $n$ edges =
number of connected diagrams with $n$ chords

Reminder

$$c_n = \sum_{k=1}^{n-1} (2k-1) c_k c_{n-k}$$
(A SAMPLE OF THE) BIJECTION PROPERTIES

We can describe an explicit bijection between bridgeless maps and connected diagrams.
(A SAMPLE OF THE) BIJECTION PROPERTIES

We can describe an explicit bijection between bridgeless maps and connected diagrams.

1) Can be extended in a bijection

maps (with bridges, or not) \[\uparrow\]

irreducible diagrams

\[\text{reducible diagram} \quad \text{irreducible diagram}\]

\{\text{already known by [Ossana de Mendez, Rosenstiehl, Cori]}\}
(A SAMPLE OF THE) BIJECTION PROPERTIES

We can describe an explicit bijection between bridgeless maps and connected diagrams.

1) Can be extended in a bijection maps (with bridges, or not) irreducible diagrams

2) Planarity characterization

Theo A map is planar iff its image under the bijection avoids the pattern
BACK TO THE TERMINAL CHORDS

diagrams $\rightarrow$ maps
terminal chords $\rightarrow$ ???
BACK TO THE TERMINAL CHORDS

diagrams $\rightarrow$ maps
terminal chords $\rightarrow$ ???

Decomposition:

$\# \text{ terminal } (C_1 \oplus C_2) = \# \text{ terminal } (C_1) + \# \text{ terminal } (C_2)$
BACK TO THE TERMINAL CHORDS

diagrams $\rightarrow$ maps
terminal chords $\rightarrow$ ???

Decomposition:

\[
\# \text{ terminal } (C_1 \oplus C_2) = \# \text{ terminal } (C_1) + \# \text{ terminal } (C_2)
\]

if $C_2 \neq \emptyset$

\[
\# \text{ terminal } (C_1 \oplus C_2) = \# \text{ terminal } (C_1)
\]

if $C_2 = \emptyset$
BACK TO THE TERMINAL CHORDS

diagrams $\rightarrow$ maps
terminal chords $\rightarrow$ ???

Decomposition:

\[
\# \quad ????\quad (C_1 \oplus C_2) \quad = \quad \# \quad ????\quad (C_1) \quad + \quad \# \quad ????\quad (C_2)
\]

if \( C_2 \neq ! \)

\[
\# \quad ????\quad (C_1 \oplus C_2) \quad = \quad \# \quad ????\quad (C_1)
\]

if \( C_2 = ! \)
BACK TO THE TERMINAL CHORDS

Diagrams $\rightarrow$ Maps
Terminal chords $\rightarrow$ ???

Decomposition:

\[
\# \text{ vertices } (C_1 \oplus C_2) = \# \text{ vertices } (C_1) + \# \text{ vertices } (C_2)
\]

if $C_2 \neq !$

\[
\# \text{ vertices } (C_1 \oplus C_2) = \# \text{ vertices } (C_1)
\]

if $C_2 = !$
Theorem

number of bridgeless maps with n edges and k vertices =

number of connected diagrams with n chords and k terminal chords
Theorem
number of bridgeless maps with n edges and k vertices = number of connected diagrams with n chords and k terminal chords

Corollary
For the uniform distribution on bridgeless combinatorial maps,

\[
\text{Number of vertices} \xrightarrow{\text{distribution}} \text{Gaussian law}
\]

mean \sim \ln(n)

variance \sim \ln(n)
PROSPECTS

→ A new bijection, mother lode of new properties
  Many open questions!

→ Combinatorial interpretation of the works of
  [Marie Yeats, Hihn Yeats]

→ Connection with lambda-calculus?
THANK YOU!