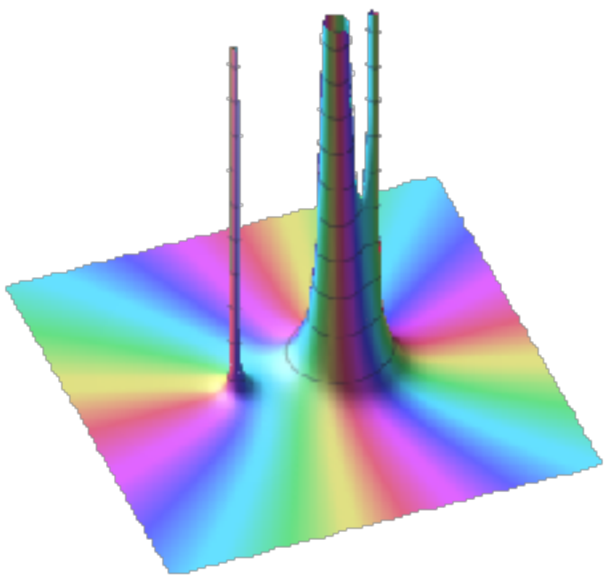
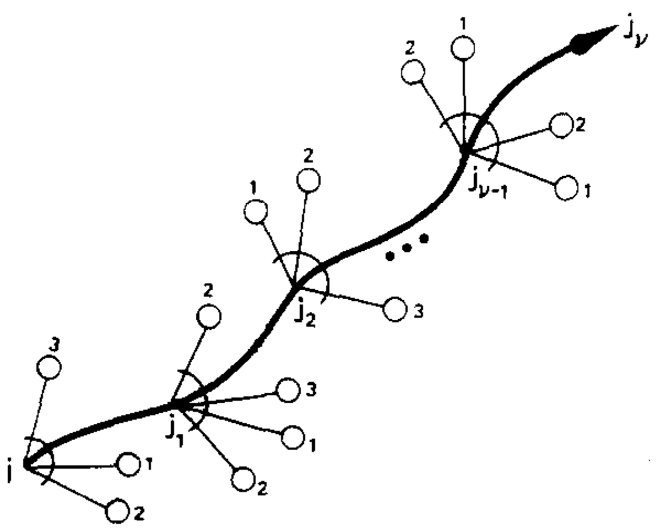
Recursive Combinatorial Structures: Enumeration, Probabilistic Analysis and Random Generation

Bruno Salvy
Inria & ENS de Lyon

Tutorial STACS2018





Combinatorics, Randomness and Analysis

From **simple local** rules, a **global** structure arises**.**

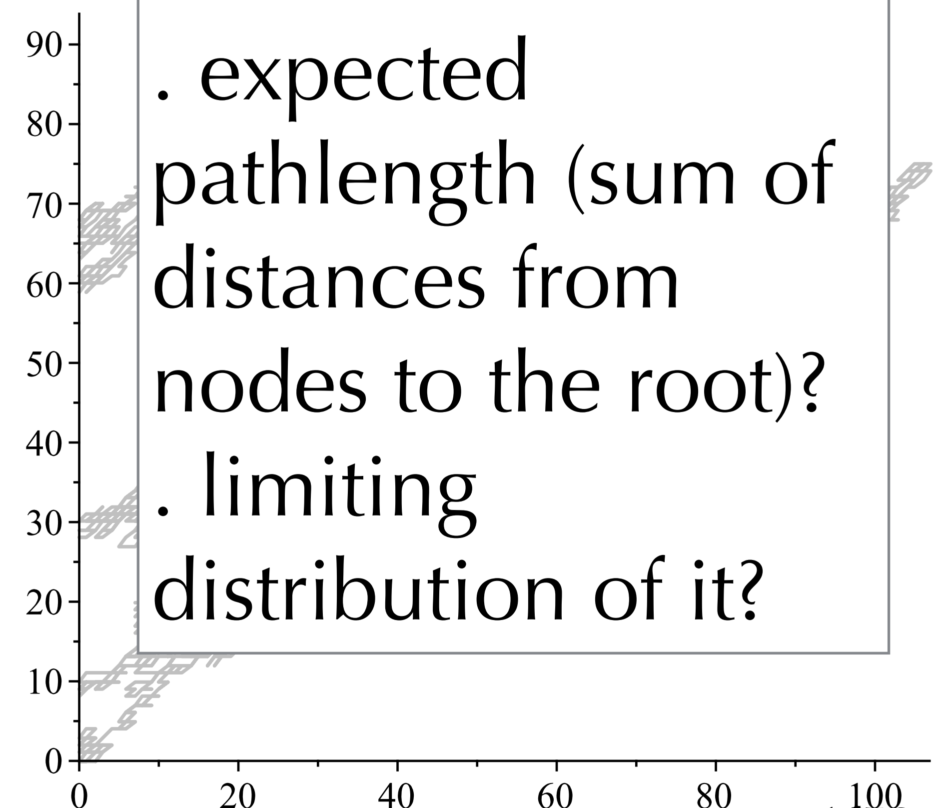
A quest for **universality** in random discrete structures**:**
➡ **probabilistic complexity** of structures and algorithms.

Quantitative results using **complex analysis.**

Typical questions

For a tree of size *n*,

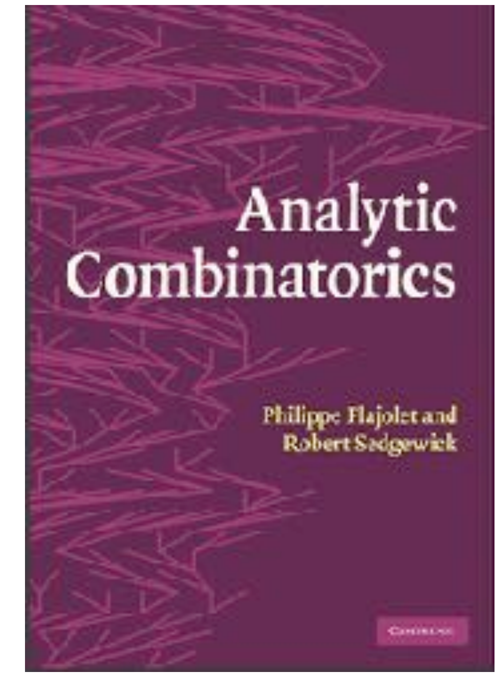
. probability that a leaf is a child of the root?
. expected pathlength (sum of distances from nodes to the root)?
. limiting distribution of it?



Philippe Flajolet, the Father of Analytic Combinatorics

1948-2011

Planned complete works span
7 volumes of approx 600 pp. each.



2009

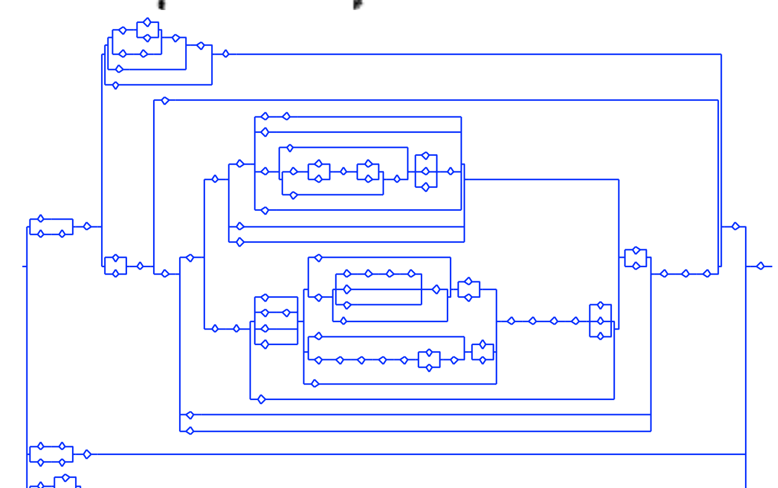
Combinatorial specification → Generating functions →

- counting;
- random generation;
- asymptotic analysis.

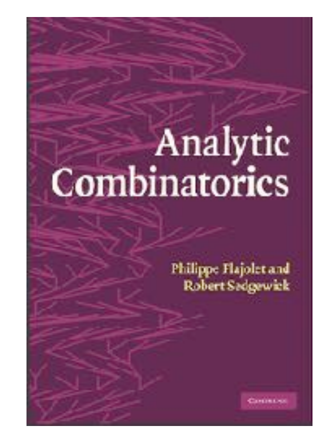
If you can specify, you can analyze.

Constructible Structures

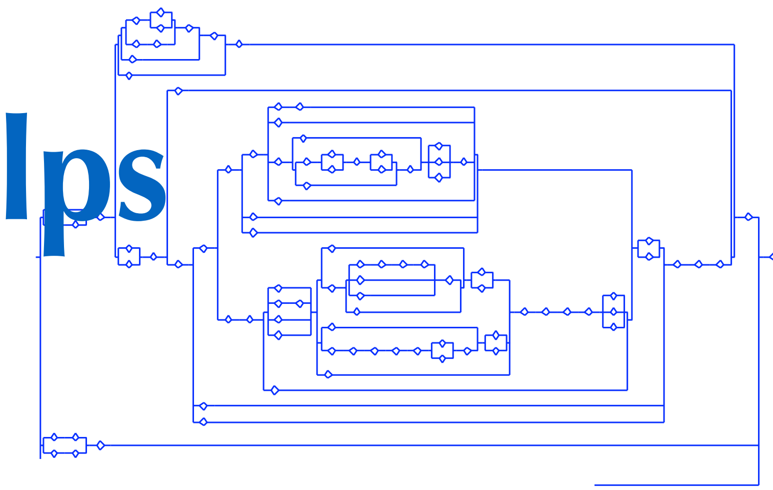
Language: 1, $\mathcal{Z}$, +, ×, SEQ, SET, CYC
and recursion.



- Binary trees: $\mathcal{B}=1+\mathcal{Z}\times\mathcal{B}\times\mathcal{B}$
- Permutations: Perm=SET(CYC($\mathcal{Z}$));
- Trees: $\mathcal{T}=\mathcal{Z}\times$SET($\mathcal{T}$);
- Functional graphs: $\mathcal{F}$=SET(CYC($\mathcal{T}$));
- Series-parallel graphs:
  $\mathcal{G}=\mathcal{Z}+\mathcal{S}+\mathcal{P}$, $\mathcal{S}=\mathrm{SEQ}_{>1}(\mathcal{Z}+\mathcal{P})$, $\mathcal{P}=\mathrm{SET}_{>1}(\mathcal{Z}+\mathcal{S})$;
- ...hundreds of examples in "the purple book".



Computer Algebra Helps



```
> with(NewtonGF):
> spgraphs:={G=Union(Z,S,P),
  S=Sequence(Union(Z,P),card>1),
  P=Set(Union(Z,S),card>1)}:
> subs(GFSeries(spgraphs,unlabelled,z,20),G);
```

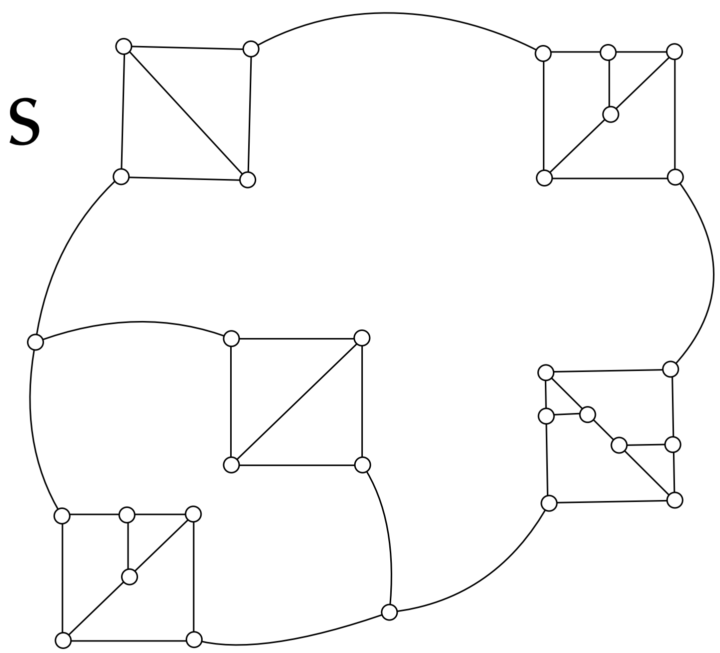
Specification

$$z + 2z^2 + 5z^3 + 15z^4 + 48z^5 + 167z^6 + 602z^7 + 2256z^8 + 8660z^9 + 33958z^{10} + 135292z^{11} + 546422z^{12} + 2231462z^{13} + 9199869z^{14} + 38237213z^{15} + 160047496z^{16} + 674034147z^{17} + 2854137769z^{18} + 12144094756z^{19} + O(z^{20})$$

Nb of distinct series-parallel graphs with 19 vertices

Bigger Specification: Planar Graphs

3-regular 2-connected Planar Graphs



A clever decomposition reduces them to interrelations between:

G_3 3-connected cores;

T, U planar triangulations (Tutte),

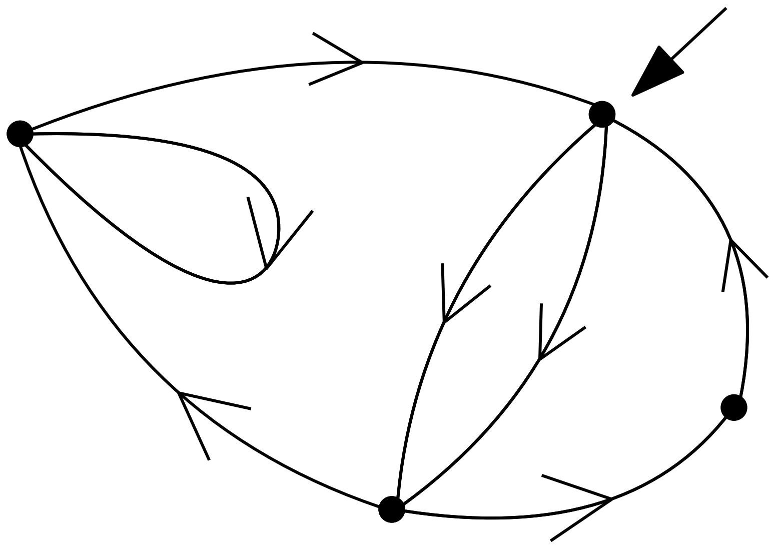
leading to

$$U = 2G_3 + T + 2U^2 = \frac{T}{(1-U)^3}, T = z(1+B)^3, B = \frac{G_3 + B^2}{1+B} + z\left(B + \frac{1}{2}B^2\right)$$

From there, many probabilistic properties follow

(e.g., proba being connected, nb triangles,...)

[Bodirsky *et al.* 2007;NoyRequiléRué2018]



Bigger Specification II: Planar Eulerian Orientations

Exponential growth still unknown

Bonichon *et alii* construct an increasing family of languages

$$\mathbb{L}^{(1)} \subset \mathbb{L}^{(2)} \subset \mathbb{L}^{(3)} \subset \cdots$$

converging to the planar Eulerian orientations.

The grammar for $\mathbb{L}^{(k)}$ contains $\approx 4^k$ equations.

Only k=1,2,3,4 can be dealt with exactly,
even with a computer!

[BonichonBousquet-MélouDorbecPennarun2017]

WANTED

Counting

Random Generation

Asymptotics

Program

15:00-now	Introduction
now-15:45	I. Counting and Generating Functions
15:45-16:15	II. Newton Iteration and Fast Algorithms
16:15-16:45	III. Algorithms for Random Generation
16:45-17:15	BREAK
17:15-18:00	IV. Algorithms for Asymptotic Estimates

I. Counting and Generating Functions

$$F(z) = \sum_{n=0}^{\infty} f_n z^n$$

Example: Binary Trees

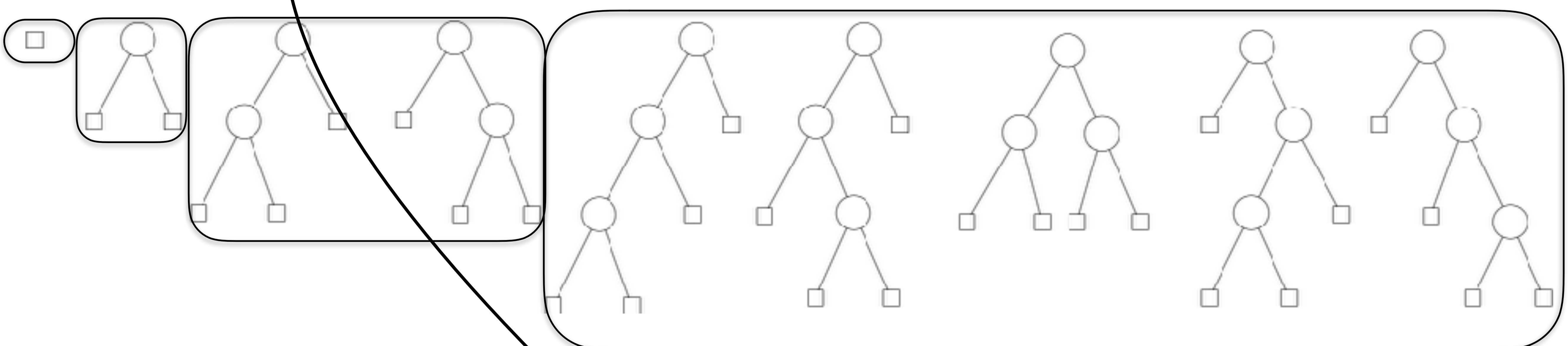
$$\mathcal{B} = 1 + \mathcal{Z} \times \mathcal{B} \times \mathcal{B}$$

$$B(z) = \sum_{n=0}^{\infty} B_n z^n$$

number of binary trees with *n* nodes

(Catalan)

$B_0 = B_1 = 1$ $B_2 = 2$ $B_3 = 5$

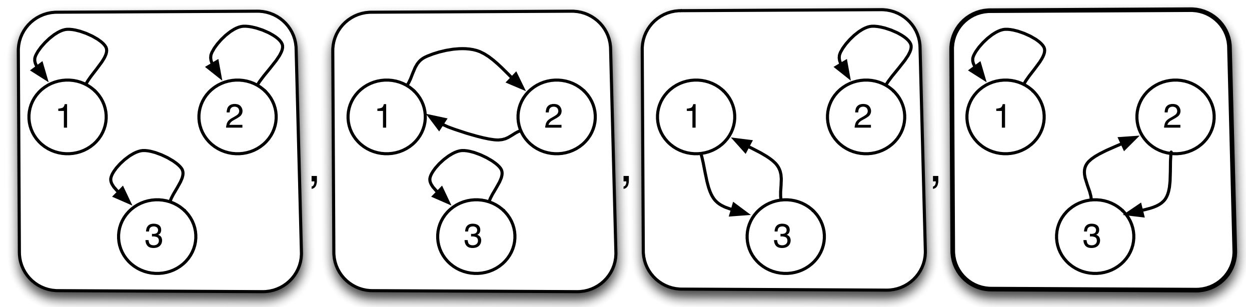


For $n \geq 1, \quad B_n = \sum_{k=0}^{n-1} B_k B_{n-k-1} \Rightarrow B_n z^n = \sum_{k=0}^{n-1} z(B_k z^k)(B_{n-k-1} z^{n-k-1})$

Demo NewtonGF

$$B(z) = 1 + zB(z)^2 = 1 + z + 2z^2 + 5z^3 + $$

What do we count?

$\mathrm{Inv}[\{1,2,3\}] = \{$  $\}$

4 involutions;

3 of them permuted by $\mathfrak{S}_3 \rightarrow$ 2 *unlabelled structures*.

Exponential generating series (EGF):

$$F(z) = \sum_{n=0}^{\infty} f_n \frac{z^n}{n!}, \quad f_n = \text{nb. labelled structs of size } n. \qquad \mathrm{Inv}_3(z) = \frac{2}{3} z^3$$

Ordinary generating series (OGF):

$$\tilde{F}(z) = \sum_{n=0}^{\infty} \tilde{f}_n z^n, \quad \tilde{f}_n = \text{nb. unlabelled structs of size } n. \qquad \widetilde{\mathrm{Inv}}_3(z) = 2z^3$$

Product Translates into Product

$$\mathcal{C} = \mathcal{A} \times \mathcal{B}$$

Unlabelled

$$\tilde{C}(z) = \sum_{n\geq 0} \tilde{c}_n z^n$$

$$= \sum_{n\geq 0} \left(\sum_{k=0}^{n} \tilde{a}_k \tilde{b}_{n-k} \right) z^n$$

$$= \sum_{n\geq 0} \sum_{k=0}^{n} (\tilde{a}_k z^k)(\tilde{b}_{n-k} z^{n-k})$$

$$\tilde{C}(z) = \tilde{A}(z) \times \tilde{B}(z)$$

Labelled

$$C(z) = \sum_{n\geq 0} c_n \frac{z^n}{n!}$$

$$= \sum_{n\geq 0} \left(\sum_{k=0}^{n} \binom{n}{k} a_k b_{n-k} \right) \frac{z^n}{n!}$$

relabelling

$$= \sum_{n\geq 0} \sum_{k=0}^{n} \left(\frac{a_k}{k!} z^k \right) \left(\frac{b_{n-k}}{(n-k)!} z^{n-k} \right)$$

$$C(z) = A(z) \times B(z)$$

Cor.: Sequences Have no Symmetry

$$\textsc{Seq}(\mathscr{A}) = 1 + \mathscr{A} + \mathscr{A} \times \mathscr{A} + \mathscr{A} \times \mathscr{A} \times \mathscr{A} + \cdots$$

translates as

$$1 + A(z) + A(z)^2 + A(z)^3 + \cdots = \frac{1}{1 - A(z)}$$

regardless of labelling.

A Dictionary for Generating Series

Language: $1, \mathcal{Z}$, +, ×, SEQ, SET, CYC and recursion.

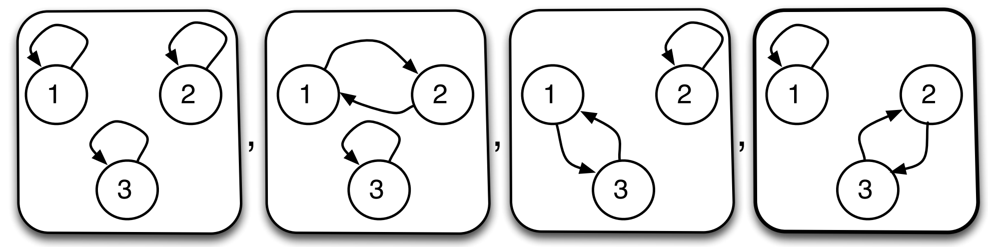
describes ***constructible structures***

Structure	EGF	OGF
$\mathcal{A}+\mathcal{B}$	$A(z)+B(z)$	$\tilde{A}(z)+\tilde{B}(z)$
$\mathcal{A}\times\mathcal{B}$	$A(z)\times B(z)$	$\tilde{A}(z)\times\tilde{B}(z)$
SEQ($\mathcal{A}$)	$\frac{1}{1-A(z)}$	$\frac{1}{1-\tilde{A}(z)}$
SET($\mathcal{A}$)	$\exp(A(z))$	$\exp(\sum\tilde{A}(z^i)/i)$
CYC($\mathcal{A}$)	$\log\frac{1}{1-A(z)}$	$\sum\frac{\phi(i)}{i}\log\frac{1}{1-\tilde{A}(z^i)}$

internal symmetry matters

$\phi(i) := \operatorname{card}\{j\in\mathbb{N}^* \mid j\le i \text{ and } \gcd(i,j)=1\}$

Example: Involutions

$\text{Inv}[\{1,2,3\}] = \Big\{$  $\Big\}$

$$\text{Inv} = \text{Set}(\text{Cyc}_{\leq 2}(\mathcal{Z}))$$

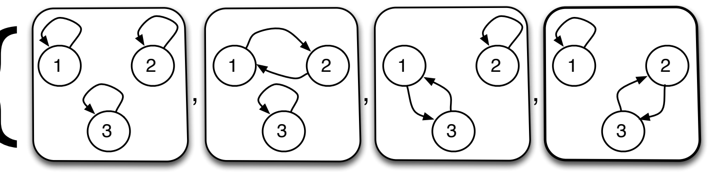
Labelled

$$\text{Inv}(z) = \exp\left(z + \frac{z^2}{2}\right)$$
$$= 1 + z + z^2 + \frac{2}{3}z^3 + \frac{5}{12}z^4 + \cdots$$

Unlabelled $\text{Cyc}_{\leq 2}(\mathcal{Z}) \mapsto z + z^2$

$$\widetilde{\text{Inv}}(z) = \exp\left(\log\frac{1}{1-z} + \log\frac{1}{1-z^2}\right) = \frac{1}{(1-z)(1-z^2)}$$
$$= 1 + z + 2z^2 + 2z^3 + 3z^4 + \cdots$$

A program can do it for you

$\mathrm{Inv}[\{1,2,3\}] = \{$  $\}$

```
> involutions:={Inv=Set(Cycle(Atom,card<=2))}:
> combstruct[gfeqns](involutions,labelled,z);
```

$$\left[Inv(z) = \mathrm{e}^{z + \frac{1}{2} z^2} \right]$$

```
> combstruct[gfeqns](involutions,unlabelled,z);
```

$$\left[Inv(z) = \mathrm{e}^{\sum_{j_1=1}^{\infty} \frac{z^{2 j_1} + z^{j_1}}{j_1}} \right]$$

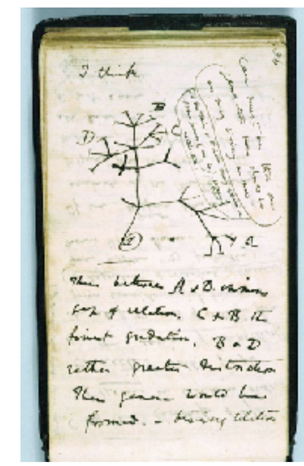
```
> simplify(value(%)) assuming z>0,z<1;
```

$$\left[Inv(z) = \frac{1}{(-1+z)^2 (z+1)} \right]$$

More Examples

Binary trees: $\mathscr{B}=1+\mathscr{Z}\times\mathscr{B}\times\mathscr{B}$

$$\longrightarrow B(z) = 1 + zB(z)^2 = \tilde{B}(z)$$



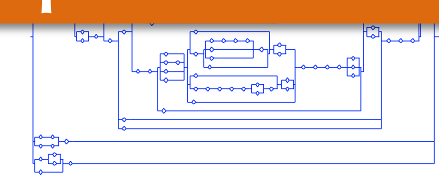
Cayley trees: $\mathscr{T}=\mathscr{Z}\times\mathrm{SET}(\mathscr{T})$

$$\longrightarrow T(z) = z\exp(T(z));$$

$$\longrightarrow \tilde{T}(z) = z\exp\left(\tilde{T}(z) + \frac{1}{2}\tilde{T}(z^2) + \frac{1}{3}\tilde{T}(z^3) + \cdots\right)$$

Next: what can we do with all these equations?

Series-parallel graphs:



$$\mathcal{G} = \mathcal{Z} + \mathcal{S} + \mathcal{P}, \mathcal{S} = \mathrm{SEQ}_{>0}(\mathcal{Z} + \mathcal{P}), \mathcal{P} = \mathrm{SET}_{>0}(\mathcal{Z} + \mathcal{S})$$

$$\longrightarrow \left\{ G(z) = z + S(z) + P(z), S(z) = \frac{1}{1 - z - P(z)} - 1, P(z) = e^{z+S(z)} - 1 \right\}$$

Ideas of Proofs for Sets & Cycles (labelled case)

k-tuples in $\underbrace{\mathcal{A} \times \cdots \times \mathcal{A}}_{k \text{ times}}$ have egf:

$$A^k(z) = \sum_{n \geq 0} t_{n,k} \frac{z^n}{n!}$$

nb. of such *k*-tuples with *n* atoms

Sets: forget the order in the *k*-tuple $\mapsto \frac{t_{n,k}}{k!} \mapsto \frac{A^k(z)}{k!}$

Cycles: forget the cyclic order in the *k*-tuple $\mapsto \frac{t_{n,k}}{k} \mapsto \frac{A^k(z)}{k}$

Ideas of Proof for Sets (unlabelled case)

1. $\mathcal{A}$ has only 1 element, *a*

$$\mathrm{UnlabSet}(\mathcal{A}) \simeq \mathrm{Seq}(a) \mapsto \frac{1}{1-z^{|a|}}$$

Notation:
$|a|$: size of *a*

2. $\mathcal{A}$ has 2 elements, *a* and *b*

$$\mathrm{UnlabSet}(\mathcal{A}) \simeq \mathrm{Seq}(a) \times \mathrm{Seq}(b) \mapsto \frac{1}{1-z^{|a|}} \times \frac{1}{1-z^{|b|}}$$

3. General case

$$\prod_{a\in\mathcal{A}} \frac{1}{1-z^{|a|}} = \exp \sum_{a\in\mathcal{A}} \underbrace{\log \frac{1}{1-z^{|a|}}}_{z^{|a|}+\frac{1}{2}z^{2|a|}+\frac{1}{3}z^{3|a|}+\cdots}$$

$$= \exp\left(A(z) + \frac{1}{2}A(z^2) + \frac{1}{3}A(z^3) + \cdots\right)$$

Cycles: inclusion-exclusion.

Irreducible Polynomials over a Finite Field

$$I(z)=\sum_{n>0} i_n z^n \qquad i_n = \text{card}\{\text{monic irred pol of deg } n \text{ over } \mathbb{F}_q\}$$

$$\boxed{\text{MonicPol} = \text{Seq}(\text{Coeffs}) = \text{Set}(\text{MonicIrred})}$$

$$P(z)=\frac{1}{1-qz}=\exp\left(I(z)+\frac{1}{2}I(z^2)+\frac{1}{3}I(z^3)+\cdots\right)$$

$$I(z)=\sum_{k>0}\frac{\mu(k)}{k}\log\frac{1}{1-qz^k} \qquad \mu(k)=\begin{cases}0 & \text{if } k \text{ has a square prime factor,}\\ (-1)^{\text{nb facts}} & \text{otherwise.}\end{cases}$$

Möbius inversion

$$=qz+\frac{q(q-1)}{2}z^2+\frac{q(q^2-1)}{3}z^3+\frac{q^2(q^2-1)}{4}z^4+\cdots$$

→ proba irred: $\frac{i_n}{q^n}\to\frac{1}{n}, \qquad n\to\infty$

→ probabilistic analysis of factoring algorithms

[FlajoletGourdonPanario2001]

Parameters

- Equations over combinatorial structures + parameters
- Multivariate generating series

$$F(z,u)=\sum_{n,k} f_{n,k}u^k z^n$$

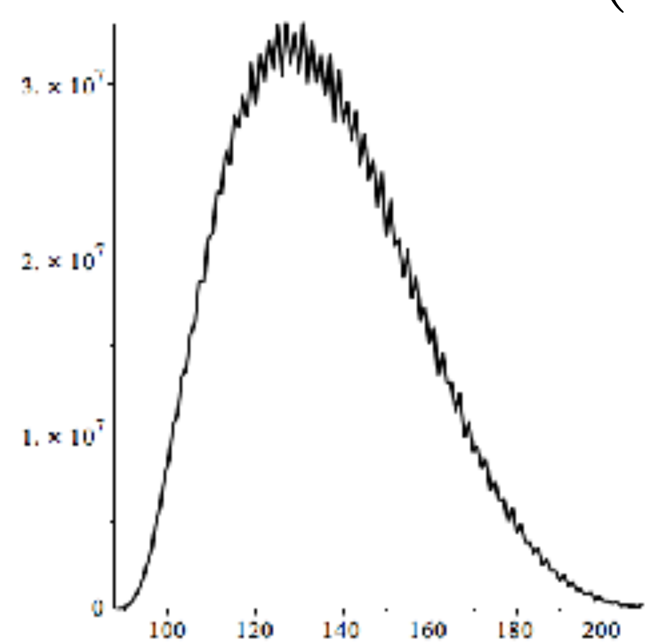
- Complex analysis
  $f_{n,k}\sim\cdots, n\to\infty$
- Limiting distribution

Ex.: path length in binary trees

$$B(z,u)=\sum_{b\in\mathcal{B}} u^{\mathrm{pl}(b)} z^{|b|}$$

$$P(z)=\left.\frac{\partial B(z,u)}{\partial u}\right|_{u=1}$$

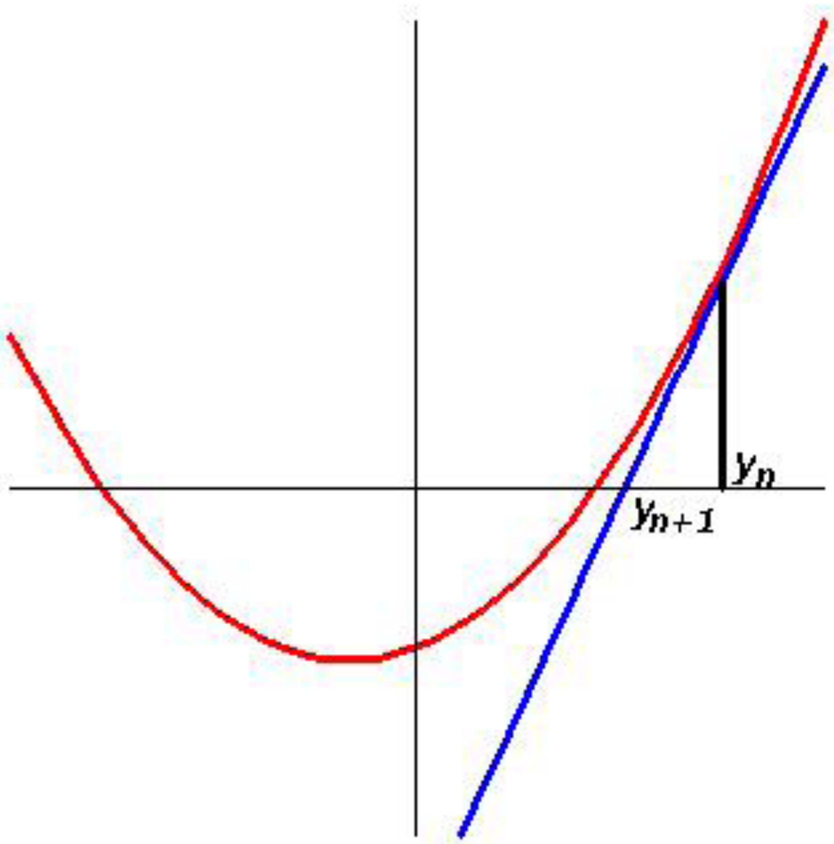
$$\frac{P_n}{nB_n}\sim\sqrt{\pi n}$$

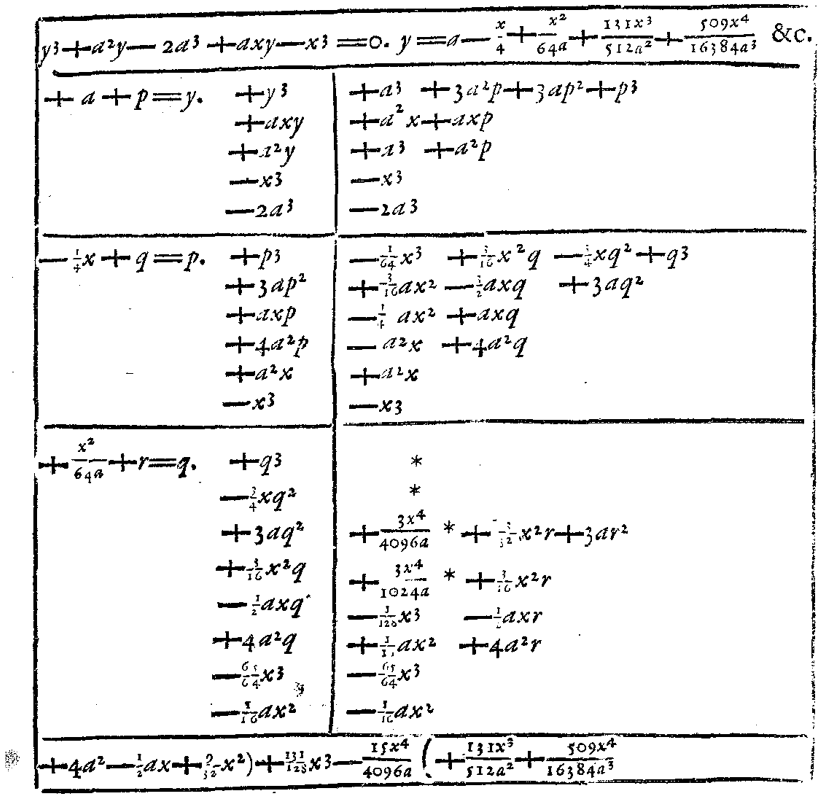


[Mishna2003]

Demo.

II. Newton Iteration and Fast Enumeration





1671

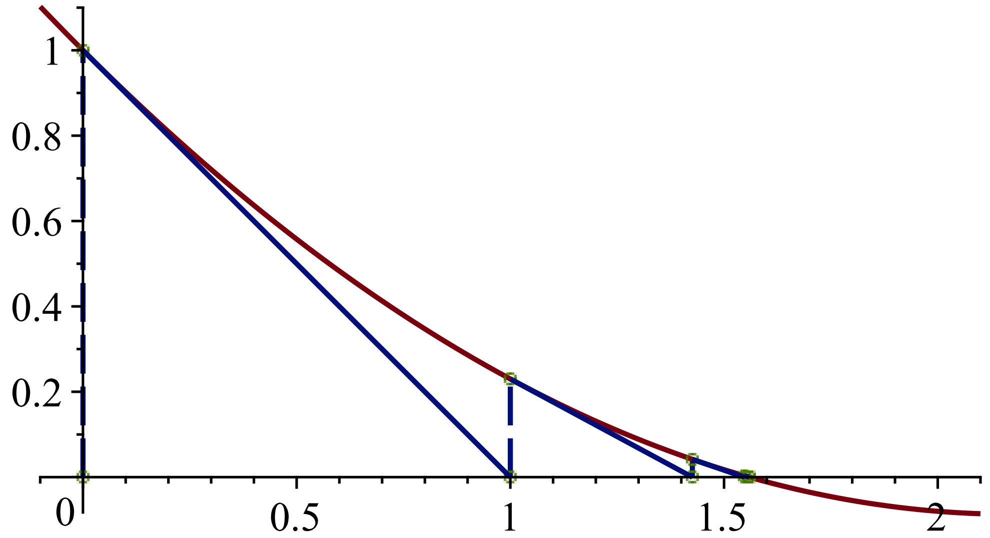
Numerical Newton Iteration

To solve $\phi(y) = 0$, iterate
$$y^{[n+1]} = y^{[n]} + u^{[n]},$$
with $\phi(y^{[n]}) + \phi'(y^{[n]})u^{[n]} = 0.$

$$\phi(y) = 1 + zy^2 - y$$

$$y^{[n+1]} = \mathcal{N}(y^{[n]}) = y^{[n]} + \frac{1 + zy^{[n]^2} - y^{[n]}}{1 - 2zy^{[n]}}$$

$$z = 0.23$$



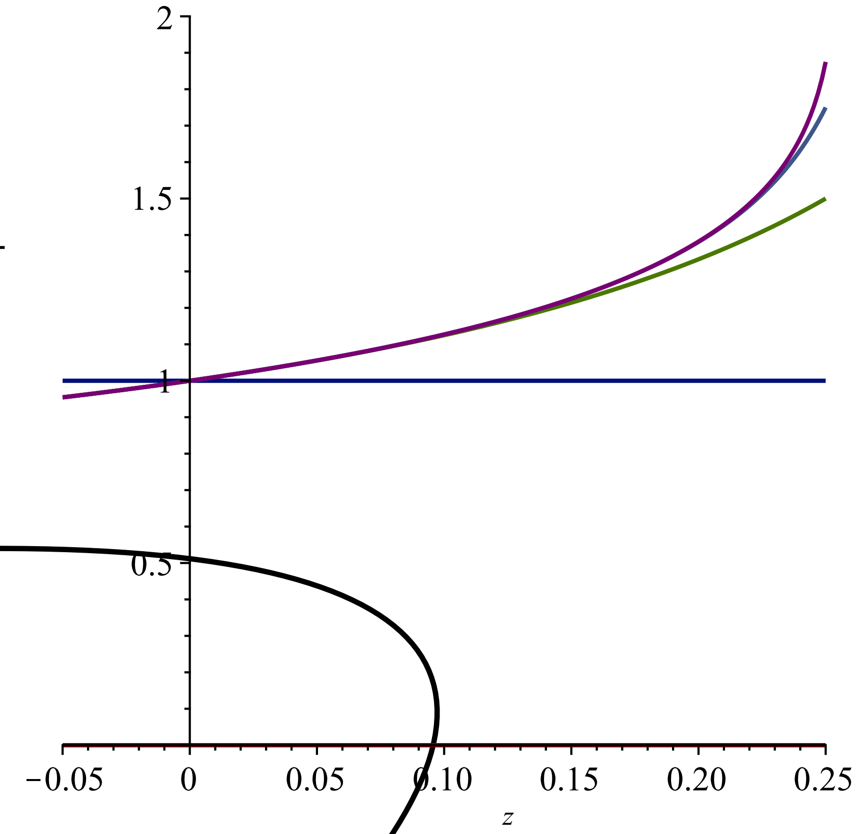
$$y^{[0]} = 0,$$
$$y^{[1]} = 1.0000000000000000,$$
$$y^{[2]} \simeq 1.425925925925925 9,$$
$$y^{[3]} \simeq 1.547193318183630 3,$$
$$y^{[4]} \simeq 1.558925660274882 2,$$
$$y^{[5]} \simeq 1.559037571392659 2,$$
$$y^{[6]} \simeq 1.559037581576915 1$$

Quadratic convergence

Newton Iteration for Power Series

Same Newton Iteration

$$y^{[n+1]} = \mathcal{N}(y^{[n]}) = y^{[n]} + \frac{1 + z{y^{[n]}}^2 - y^{[n]}}{1 - 2zy^{[n]}}$$



$$y^{[0]} = 0$$

$$y^{[1]} = 1$$

$$y^{[2]} = 1 + z + 2z^2$$

$$y^{[3]} = 1 + z + 2z^2 + 5z^3 + 14z^4 + 42z^5 + 132z^6$$

Proving numerical convergence requires control over the tails

On power series: $y - y^{[\infty]} = O(z^{m}) \Longrightarrow \mathcal{N}(y) - y^{[\infty]} = O(z^{2m(+1)})$

```
Expand(N) = {
   res=Expand(N/2);
   a=ϕ(res); b=ϕ'(res);
   u=Solve(a+bx,x);
   return res+u; }
```

Cost(N) ≤ ct × Cost(last step)

Example: Newton Iteration for Inverses

To solve $\phi(y) = 0$, iterate
$$y^{[n+1]} = y^{[n]} + u^{[n]},$$
with $\phi(y^{[n]}) + \phi'(y^{[n]})u^{[n]} = 0.$

$$\phi(y) = a - 1/y$$
$$\Rightarrow 1/\phi'(y) = y^2$$
$$\Rightarrow y^{[n+1]} = y^{[n]} - y^{[n]}(ay^{[n]} - 1).$$

Cost: a small number of multiplications. Works for:

numerical inversion;
reciprocal of power series;
inversion of matrices.

[Schulz1933;Sieveking1972;Kung1974]

Series-Parallel Graphs

To solve $\phi(y) = 0$, iterate

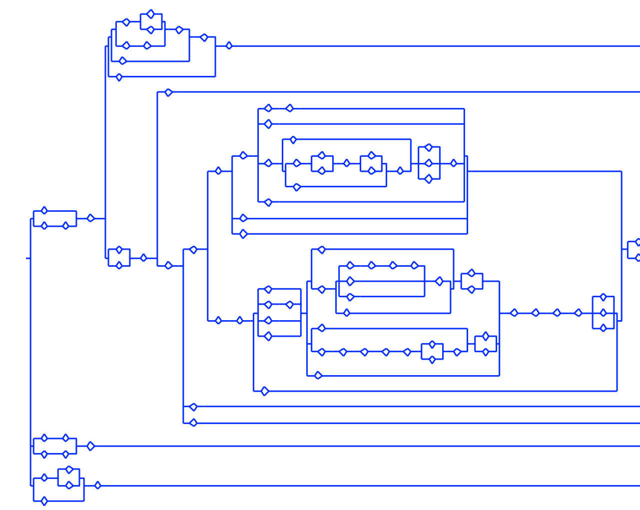
$$y^{[n+1]} = y^{[n]} + u^{[n]},$$

with $\phi(y^{[n]}) + \phi'(y^{[n]})u^{[n]} = 0.$

$$\mathcal{G} = \mathcal{Z} + \mathcal{S} + \mathcal{P},$$

$$\mathcal{S} = \mathrm{SEQ}_{>1}(\mathcal{Z} + \mathcal{P})$$

$$\mathcal{P} = \mathrm{SET}_{>1}(\mathcal{Z} + \mathcal{S})$$



Jacobian matrix

$$Y = \begin{pmatrix} G \\ S \\ P \end{pmatrix}, \quad \Phi(Y) = Y - H(Y), \quad H(Y) = \begin{pmatrix} z + S + P \\ \frac{1}{1-z-P} - 1 - z - P \\ e^{z+S} - 1 - z - S \end{pmatrix}, \quad \frac{\partial H}{\partial Y} = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & \frac{1}{(1-z-P)^2} - 1 \\ 0 & e^{z+S} - 1 & 0 \end{pmatrix}$$

$$Y^{[n+1]} = Y^{[n]} + \left(\mathrm{Id} - \frac{\partial H}{\partial Y} \right)^{-1} \cdot \left(H(Y^{[n]}) - Y^{[n]} \right)$$

fast matrix inverse by previous iteration

Next: fast exponential and logarithm (sets & cycles)

From the Inverse to the Exponential

1. Logarithm of power series: $\log f = \int (f'/f)$
2. Exponential of power series: $\phi(y) = a - \log y$

$$e^{[n+1]} = e^{[n]} + \frac{a - \log e^{[n]}}{1/e^{[n]}},$$
$$= e^{[n]} + e^{[n]}\left(a - \int e^{[n]\prime}/e^{[n]}\right).$$

$1/e^{[n]}$ is computed by Newton iteration too!

Extends to more general systems of differential equations.

[Brent1976;HanrotQuerciaZimmermann2004;Bostan *et alii* 2007]

Conclusion for Series-Parallel Graphs

$$\mathcal{G}=\mathcal{Z}+\mathcal{S}+\mathcal{P},\ \mathcal{S}=\textsc{Seq}_{>1}(\mathcal{Z}+\mathcal{P}),\ \mathcal{P}=\textsc{Set}_{>1}(\mathcal{Z}+\mathcal{S})$$

compiles into the Newton iteration:

$$\begin{cases} i^{[n+1]} = i^{[n]}(e^{[n]}i^{[n]} - 1), \\ e^{[n+1]} = e^{[n]} + e^{[n]}(z + S^{[n]} - \int(\frac{d}{dz}e^{[n]})i^{[n]}), \\ v^{[n+1]} = v^{[n]} - v^{[n]}((1 - z - P^{[n]})v^{[n]} - 1), \\ U^{[n+1]} = U^{[n]} + U^{[n]} \cdot \left(\begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & {v^{[n+1]}}^2 - 1 \\ 0 & e^{[n+1]} - 1 & 0 \end{pmatrix} \cdot U^{[n]} + \mathrm{Id} - U^{[n]} \right), \\ \begin{pmatrix} G^{[n+1]} \\ S^{[n+1]} \\ P^{[n+1]} \end{pmatrix} = \begin{pmatrix} G^{[n]} \\ S^{[n]} \\ P^{[n]} \end{pmatrix} + U^{[n+1]} \cdot \begin{pmatrix} z + S^{[n]} + P^{[n]} - G^{[n]} \\ v^{[n+1]} - 1 - z - P^{[n]} - S^{[n]} \\ e^{[n+1]} - 1 - z - S^{[n]} - P^{[n]} \end{pmatrix} \bmod z^{2^{n+1}}. \end{cases}$$

Computation reduced to products and linear operations.

Fast Enumeration

Thm. First N coefficients of GFs of all *constructible* structures:

1. arithmetic complexity $O(N \log N)$ (both ogf & egf);

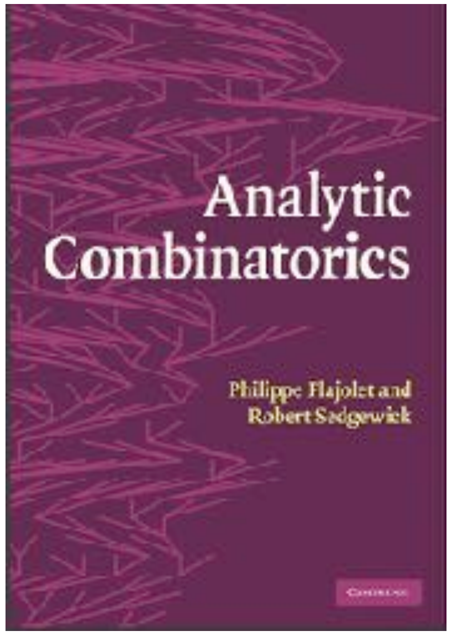
2. bit complexity

$O(N^2 \log^2 N \log\log N)$ (ogf); $O(N^2 \log^3 N \log\log N)$ (egf).

Ingredients: Newton iteration & FFT.

```
> with(NewtonGF):
> BinTrees:={B=Union(Epsilon,Prod(Z,B,B))}:
> GFSeries(BinTrees,labelled,z,21)[2];
```

$$B = 1 + z + 2z^2 + 5z^3 + 14z^4 + 42z^5 + 132z^6 + 429z^7 + 1430z^8 + 4862z^9 + 16796z^{10} + 58786z^{11} + 208012z^{12} + 742900z^{13} + 2674440z^{14} + 9694845z^{15} + 35357670z^{16} + 129644790z^{17} + 477638700z^{18} + 1767263190z^{19} + 6564120420z^{20} + O(z^{21})$$

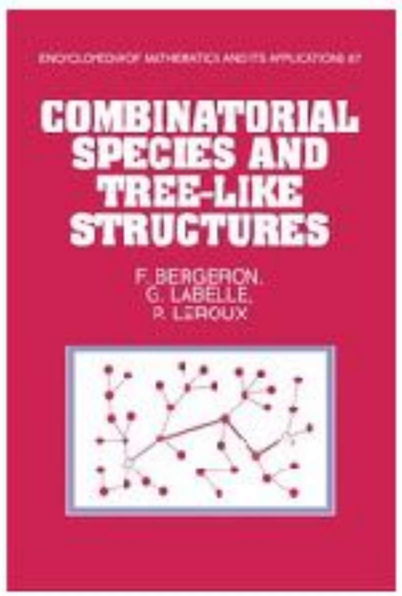


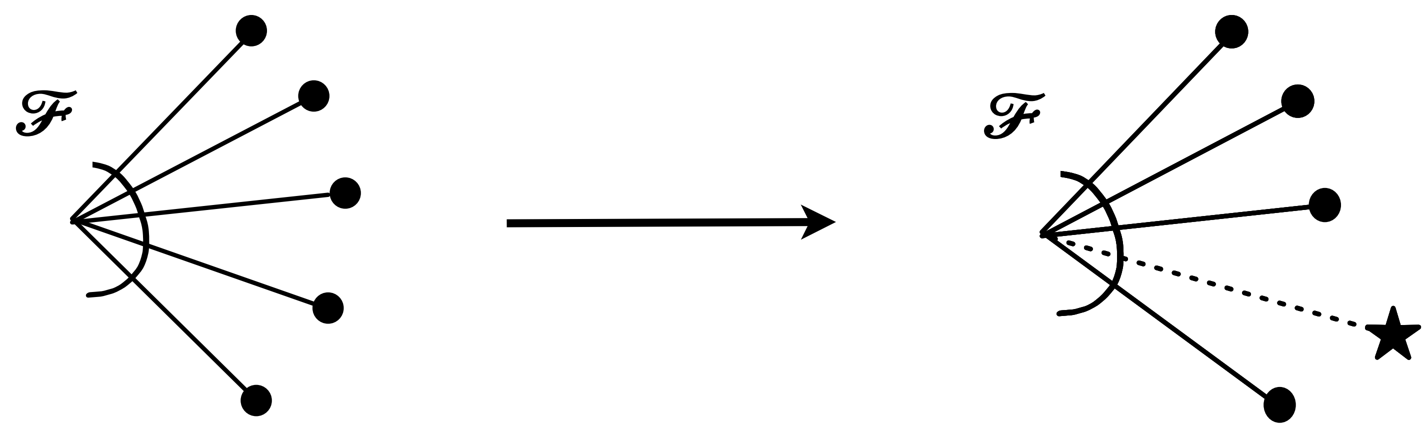
Demo NewtonGF

[Pivoteau-S-Soria 2012]

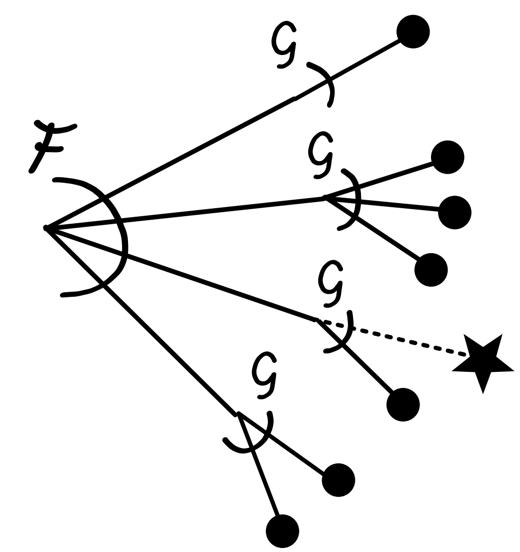
Derivative of Combinatorial Structures

(origin: species theory)





- $(\mathcal{F}+\mathcal{G})' = \mathcal{F}'+\mathcal{G}'; (\mathcal{F}\times\mathcal{G})' = \mathcal{F}'\times\mathcal{G}+\mathcal{F}\times\mathcal{G}';$
- $\mathcal{F}(\mathcal{G})' = \mathcal{F}'(\mathcal{G})\times\mathcal{G}';$
- $0' = 1' = 0;\ \mathcal{Z}' = 1;$
  $\textsc{Set}' = \textsc{Set};\ \textsc{Cyc}' = \textsc{Seq};\ \textsc{Seq}' = \textsc{Seq}\times\textsc{Seq}.$



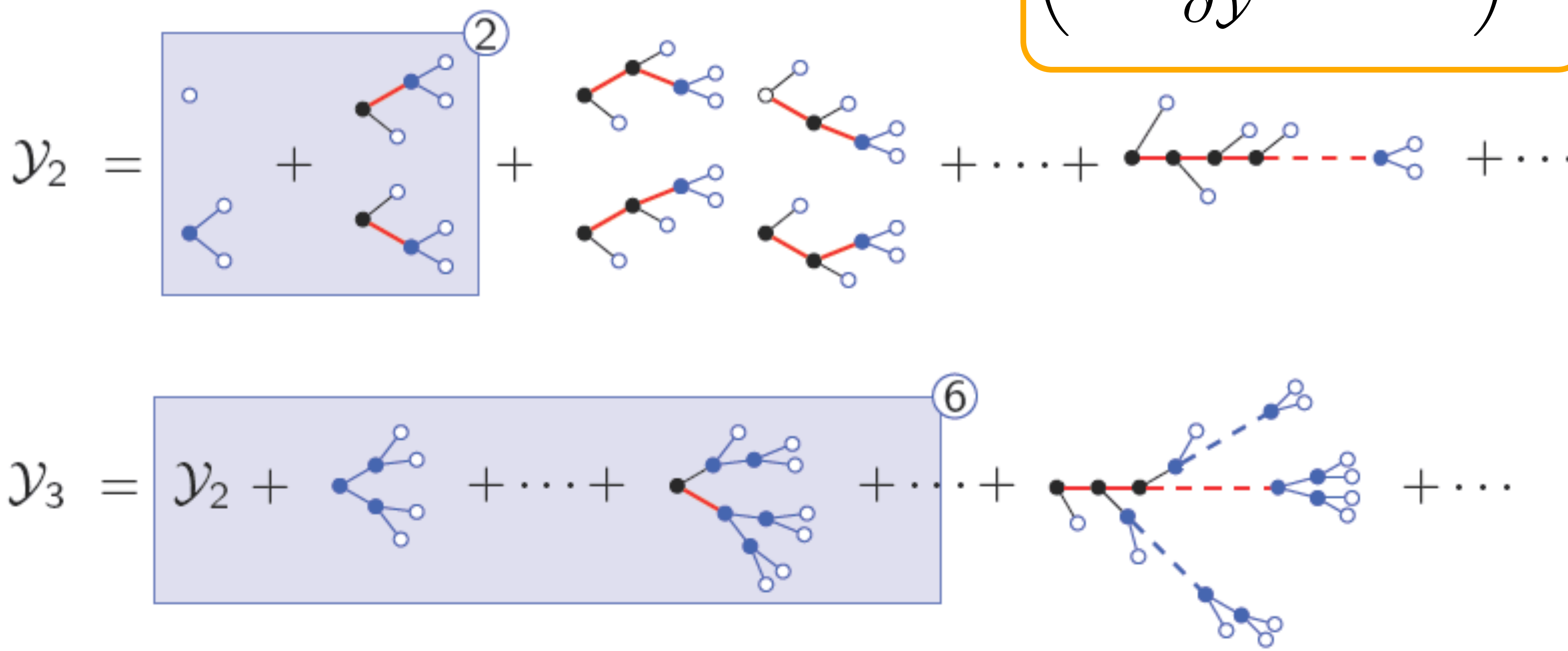
Combinatorial Newton Iteration

$$\mathcal{Y} = 1 + \mathcal{Z} \times \mathcal{Y} \times \mathcal{Y} =: \mathcal{H}(\mathcal{Z}, \mathcal{Y})$$

$$\mathcal{Y}^{[n+1]} = \mathcal{Y}^{[n]} + \textsc{Seq}(\mathcal{Z} \cdot \mathcal{Y}^{[n]} \cdot \star + \mathcal{Z} \cdot \star \cdot \mathcal{Y}^{[n]}) \cdot ((1 + \mathcal{Z} \cdot \mathcal{Y}^{[n]} \cdot \mathcal{Y}^{[n]}) \setminus \mathcal{Y}^{[n]}).$$

$$\left(\mathrm{Id} - \frac{\partial \mathcal{H}}{\partial \mathcal{Y}}(\mathcal{Z}, \mathcal{Y}^{[n]})\right)^{-1}$$

$\mathcal{Y}_0 = \varnothing \qquad \mathcal{Y}_1 = \circ$



Ccl: numerical convergence of the Newton iteration starting from 0.

[Décoste, Labelle, Leroux 1982]

General Combinatorial Newton Iteration

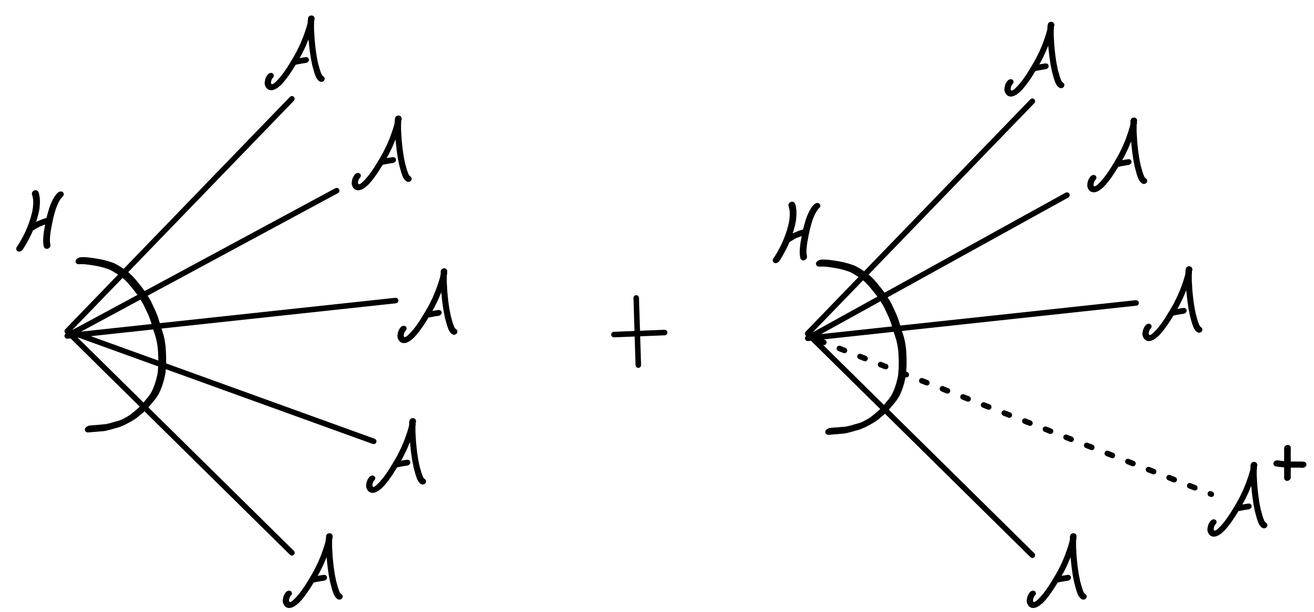
Thm. [essentially Labelle] If $\mathscr{A}$ coincides with the solution of $\mathscr{Y}=\mathscr{H}(\mathcal{Z},\mathscr{Y})$ up to size k and $\mathscr{A}\subset\mathscr{H}(\mathcal{Z},\mathscr{A})$, then

$$\mathscr{N}(\mathscr{A}):=\mathscr{A}+\sum_{i\geq 0}(\partial\mathscr{H}/\partial\mathscr{Y}(\mathcal{Z},\mathscr{A}))^i\cdot(\mathscr{H}(\mathcal{Z},\mathscr{A})\setminus\mathscr{A})$$

coincides with the solution up to size $2k+1$.

Proof. $\mathscr{N}(\mathscr{A}):=\mathscr{A}+\mathscr{A}^+$ with

$\mathscr{A}+\mathscr{A}^+ =$

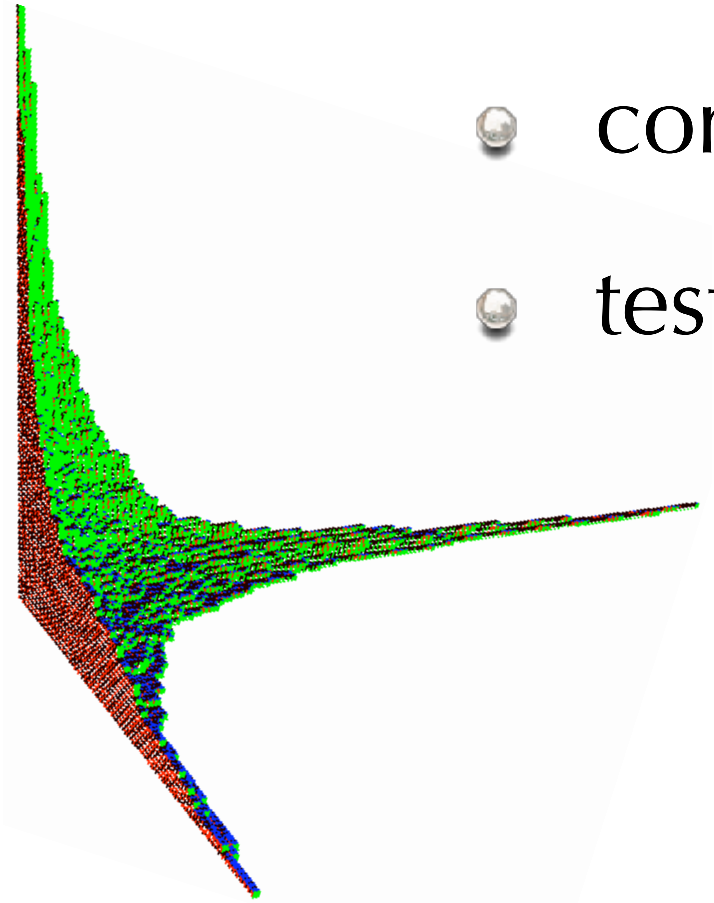


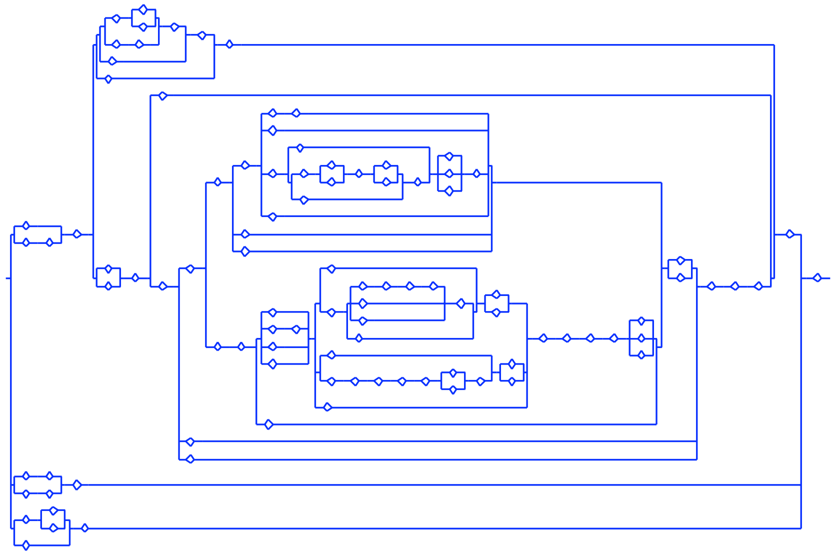
III. Random Generation

Why Random Generation?

Simulation in the discrete world; helps

- evaluate parameters;
- compare/refine models;
- test software.





Recursive Method

$$\mathcal{B} = \mathcal{Z} + \mathcal{Z} \times \mathcal{B} \times \mathcal{B}$$

```
DrawBinTree(n) = {
  if n=1 return Z;
  U:=Uniform([0,1]); k:=0; S:=0;
  while (S<U) {k:=k+1;S:=S+bkbn-k-1/bn;}
  return ZxDrawBinTree(k)xDrawBinTree(n-k-1); }
```

b_k: nb binary trees with k nodes (Catalan)

Requires $b_0, b_1, \ldots, b_n$.

[Nijenhuis & Wilf 1978; Flajolet, Zimmermann & Van Cutsem 1993-4]

General Algorithm & Complexity

```
RecGenerate(h∈𝓗;n∈ℕ) = {
  if 𝓗=1 and n=0 return 1;
  if 𝓗=𝓩 and n=1 return 𝓩;
  if 𝓗=𝓕x𝓖 {
     U:=Uniform([0,1]); k:=0; S:=0;
     while (S<U) {k:=k+1;S:=S+f_k g_{n-k}/h_n;}
     RecGenerate(f∈𝓕,k);
     RecGenerate(g∈𝓖,n-k);
     return (f,g); }
  if 𝓗=𝓕+𝓖 {
     U:=Uniform([0,1]);
     if u<f_n/h_n return RecGenerate(f∈𝓕,n);
     else return RecGenerate(g∈𝓖,n); }
  if 𝓗=Set(𝓕) {
     …
```

Main cost in this loop.
Worst-case $O(n^2)$
reduces to **O(*n* log *n*)**
by reorganization:
$(k=0,n,1,n-1,2,n-2,\ldots)$.

+ Precomputation
enumeration sequences
in ≈ ***n*²** binary ops.

Demo
Combstruct

Boltzmann Sampling

$$\text{Proba}(t) = \frac{x^{|t|}}{T(x)} \quad \text{with} \quad T(x) = \sum_{t \in \mathcal{T}} x^{|t|} = \sum_n T_n x^n$$

x is a parameter of the algorithm

```
Generate(h∈𝓗) = {
   if 𝓗=1 return 1;
   if 𝓗=𝓩 return 𝓩;
   if 𝓗=𝓕x𝓖 {
      Generate(f∈𝓕);
      Generate(g∈𝓖);
      return (f,g); }
   if 𝓗=𝓕+𝓖 {
      b:=Bernoulli(F(x)/H(x));
      if b=1 return Generate(f∈𝓕);
      else return Generate(g∈𝓖); }
   if 𝓗=Set(𝓕) {
      …
```

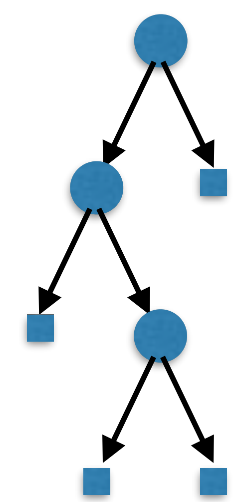
F(x),H(x) by Newton iteration (once)

Example: binary trees

$\mathcal{B} = 1 + \mathcal{Z} \times \mathcal{B} \times \mathcal{B}$

with 1/B(.23)≃1/1.559≃.6414

Coin: 0,0,1,0,1,1,1,1,0,1,…



Exp. size

$$\sim \frac{1}{2\sqrt{1-4x}}$$

[Duchon, Flajolet, Louchard, Schaeffer 2002-04]

Proof & Complexity

$$\text{Proba}(t) = \frac{x^{|t|}}{T(x)} \quad \text{with} \quad T(x) = \sum_{t \in \mathcal{T}} x^{|t|} = \sum_n T_n x^n$$

```
Generate(h∈ℋ) = {
   if ℋ=1 return 1;
   if ℋ=𝒵 return 𝒵;
   if ℋ=ℱxℊ {
      Generate(f∈ℱ);
      Generate(g∈𝒢);
      return (f,g); }
   if ℋ=ℱ+𝒢 {
      b:=Bernoulli(F(x)/H(x));
      if b=1 return Generate(f∈ℱ);
      else return Generate(g∈𝒢);
   if ℋ=Set(ℱ) {
      …
```

$$\frac{x^{|(f,g)|}}{H(x)} = \frac{x^{|f|+|g|}}{F(x)G(x)} = \frac{x^{|f|}}{F(x)} \frac{x^{|g|}}{G(x)}$$

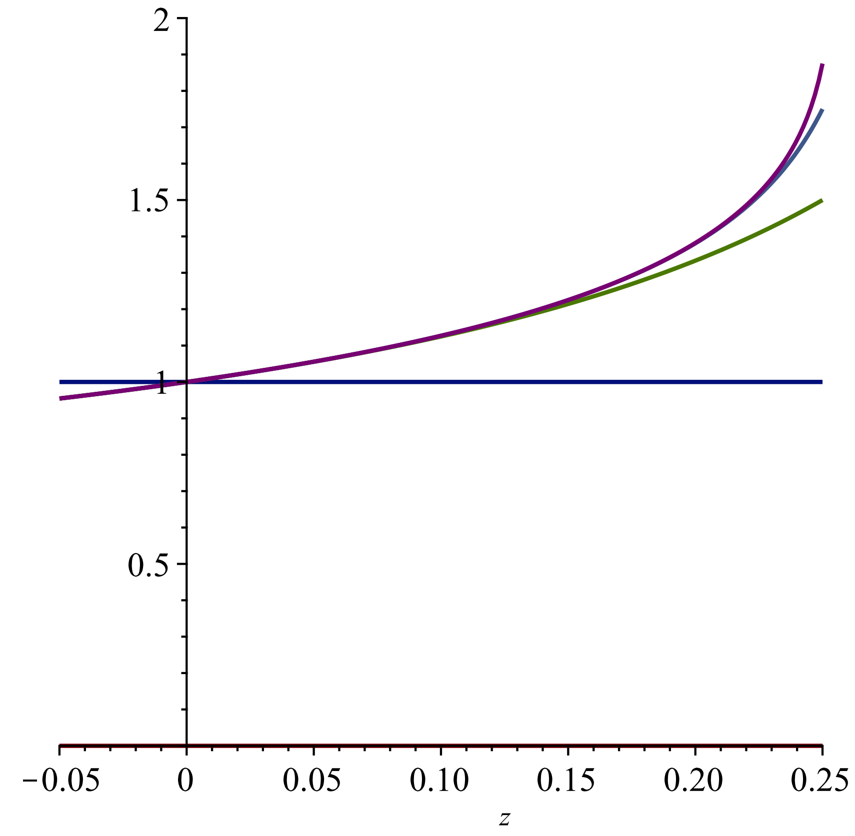
$$\frac{x^{|h|}}{H(x)} = \begin{cases} \frac{F(x)}{H(x)} \frac{x^{|h|}}{F(x)}, & \text{if } h \in \mathcal{F} \\ \frac{G(x)}{H(x)} \frac{x^{|h|}}{G(x)}, & \text{if } h \in \mathcal{G}. \end{cases}$$

Complexity: number of arith. ops **linear** in $|t|$.

[Duchon, Flajolet, Louchard, Schaeffer 2002-04]

Example: XML-trees

Grammar	File size	#eq.	Newton
rss	9.5k	16	0.02s.
Relax NG	124k	114	0.10s.
XSLT	168k	122	0.12s.
XHTML Basic	284k	96	0.32s.
SVG	6.3M	232	0.23s.
OpenDocument	2.8M	814	0.34s.
DocBook	11M	977	23s.



Time for *x* s.t. E(size)=10,000

Demo Combstruct

[Darrasse2010]

Targeting an Approximate Size

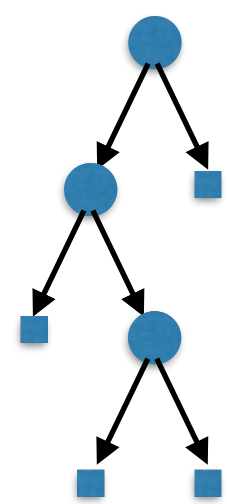
$$\text{Proba}(t) = \frac{x^{|t|}}{T(x)} \quad \text{with} \quad T(x) = \sum_{t \in \mathcal{T}} x^{|t|}$$

Expected size and standard deviation:

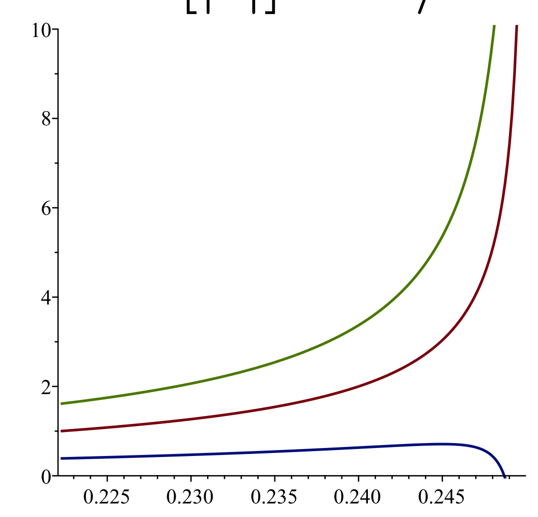
$$\mathbb{E}[|t|] = \frac{xT'(x)}{T(x)} \qquad \mathbb{E}[|t|^2] = \frac{x^2T''(x) + xT'(x)}{T(x)} \qquad \sigma = \sqrt{\mathbb{E}[|t|^2] - \mathbb{E}[|t|]^2}$$

Binary trees

$\mathcal{B} = 1 + \mathcal{Z} \times \mathcal{B} \times \mathcal{B}$

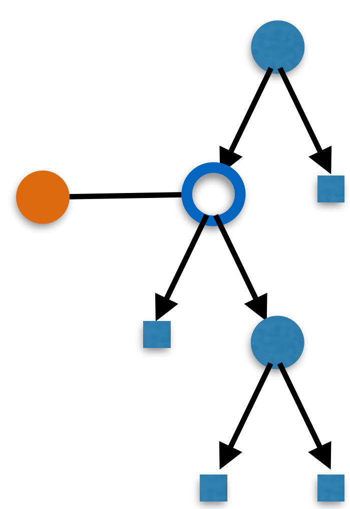


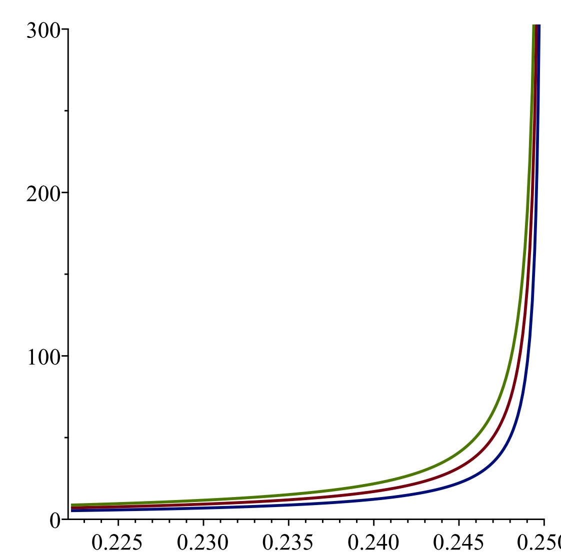
$\mathbb{E}[|t|] \pm \sigma/4$



Pointed binary trees

$\mathcal{Z} \times \mathcal{B}'$





Rejection

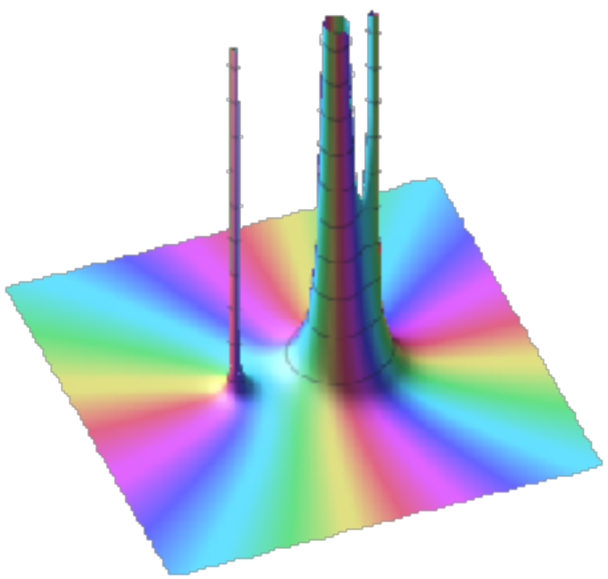
```
1. Find (once) x s.t. E(|t|)=n
2. Run Generate until |t|∈(n(1-ε),n(1+ε)).
```

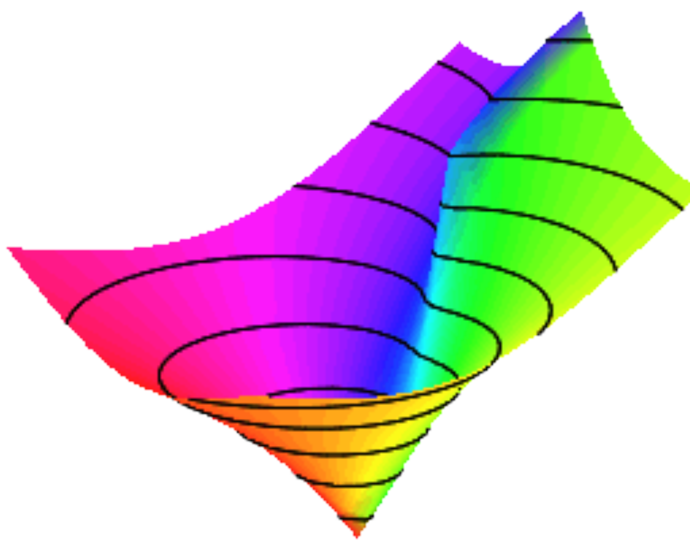
Nicest cases: standard deviation/expected size —> 0
More common: ratio has a finite limit
Otherwise: use pointing.

For all constructible structures, approximate size n can be reached in O(n) arith op.

Choice of strategy relies on singularity analysis (after the break).

IV. Algorithms for Asymptotic Analysis





$$F(z) = \sum_{n=0}^{\infty} f_n z^n \longrightarrow f_n \sim \cdots, \quad n \to \infty$$

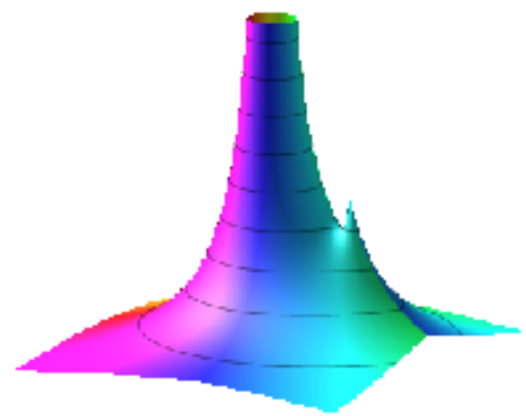
Singularity Analysis

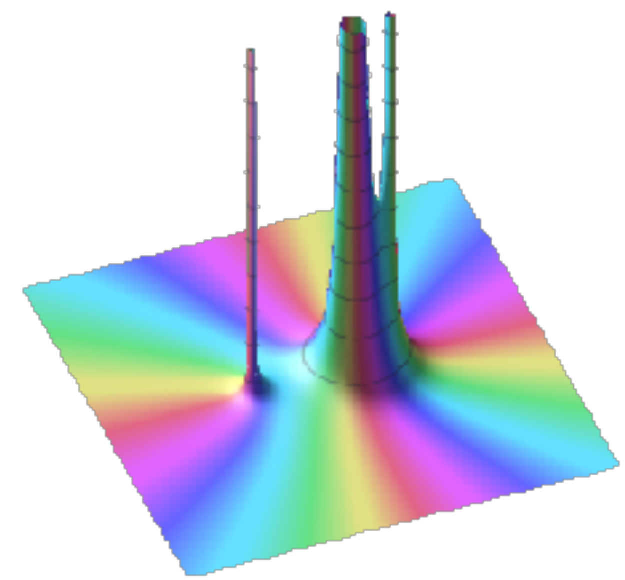
counts the number of objects of size n → $(a_n) \mapsto A(z) := \sum_{n\geq 0} a_n z^n$ ← captures some structure

A 3-Step Method:

1. Locate dominant singularities
2. Compute local behaviour
3. Translate into asymptotics

$$a_n = \frac{1}{2\pi i}\oint \frac{A(z)}{z^{n+1}}\,dz$$





$$A(z) \underset{z\to\rho}{\sim} c\left(1-\frac{z}{\rho}\right)^{\alpha} \log^m \frac{1}{1-\frac{z}{\rho}} \qquad a_n \underset{n\to\infty}{\sim} c\,\rho^{-n}\frac{n^{-\alpha-1}}{\Gamma(-\alpha)}\log^m n$$

$(\alpha \notin \mathbb{N})$

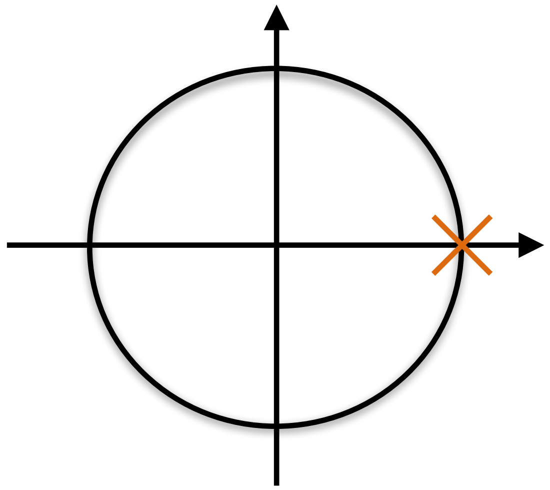
full asymptotic expansion available

[FlajoletOdlyzko1990]

Next: computation of ρ, α, m, c

Pringsheim's Theorem

$$(a_n) \mapsto A(z) := \sum_{n \geq 0} a_n z^n$$



Useful property [Pringsheim Borel]

$a_n \geq 0$ for all $n \Longrightarrow$ real positive dominant singularity.

A couple of years after the publication of my Thesis, Edmund Landau wrote to me to say that the German mathematician Pringsheim thought he had discovered the theorem [...] ``But'', he added, ``of course, this result is yours''. [...] what would have happened if I had not been alive to receive his letter? He would not have been undeceived and the discovery would have been thought to be mine—until other readers would have restored it, not to Pringsheim but to E. Borel.

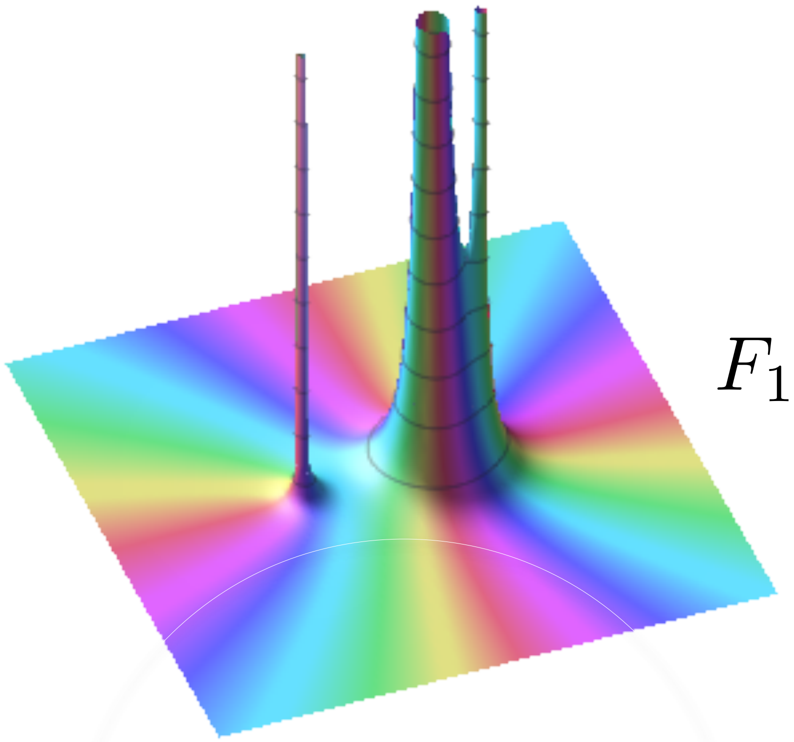
J. Hadamard (1954)

IV. 1. Rational Generating Functions

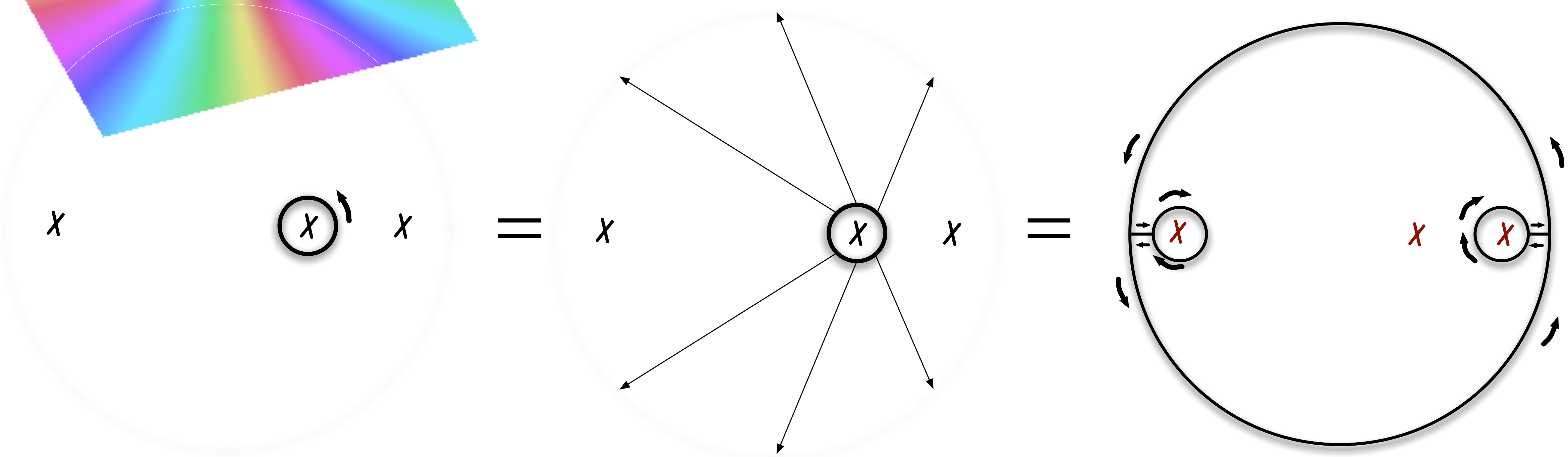
Linear Recurrences with Constant Coefficients

Fibonacci Numbers

$$a_n = \frac{1}{2\pi i}\int_\gamma \frac{f(z)}{z^{n+1}}dz$$



$$F_1 = 1 = \frac{1}{2\pi i}\oint \frac{1}{1-z-z^2}\frac{dz}{z^2}$$



As n increases, the smallest singularities dominate.

$$F_n = \frac{\phi^{-n-1}}{1+2\phi} + \frac{\overline{\phi}^{-n-1}}{1+2\overline{\phi}}$$

Conway's sequence

1,11,21,1211,111221,...

Generating function for lengths:

$$f(z)=P(z)/Q(z)$$

with deg Q=72.

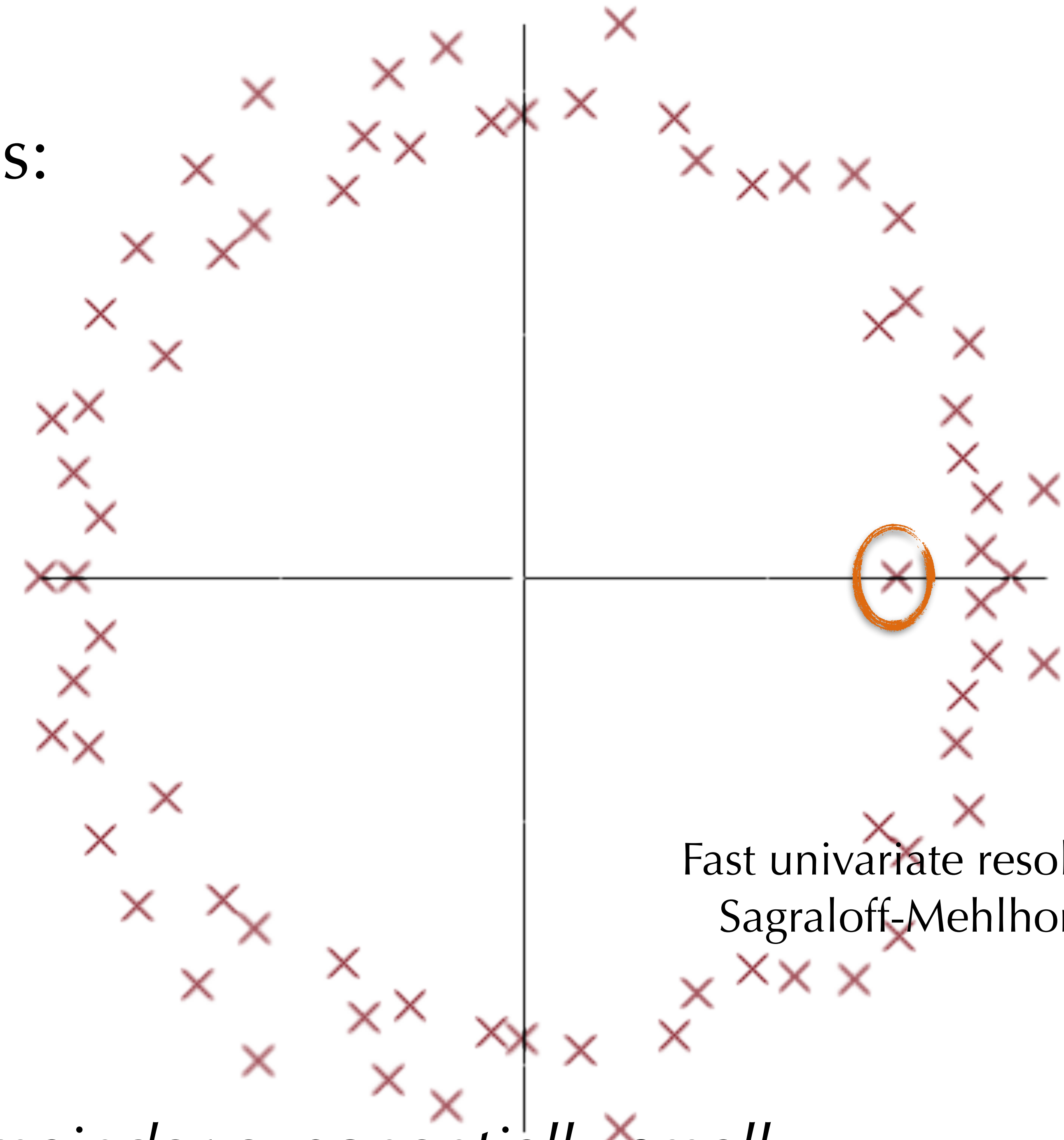
Smallest singularity:

$\rho \simeq 0.7671198507$

$$\ell_n \simeq 2.04216\, \rho^{-n}$$

$c=\rho^{-1}\,\mathrm{Res}(f,\rho)$

algebraic



Fast univariate resolution: Sagraloff-Mehlhorn16

remainder exponentially small

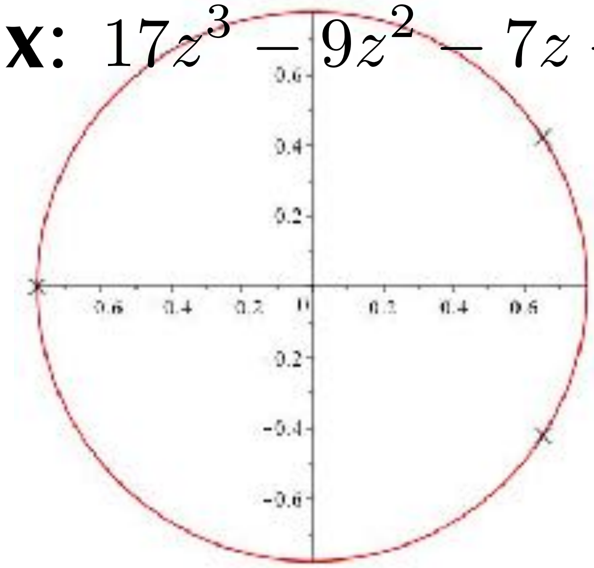
[Conway 1987]

Singularity Analysis for Rational Functions

A 3-Step Method:

1. Locate dominant singularities
   a. singularities; b. dominant ones
2. Compute local behaviour
3. Translate into asymptotics

Ex: $17z^3 - 9z^2 - 7z + 8$



dist 10^{-5}

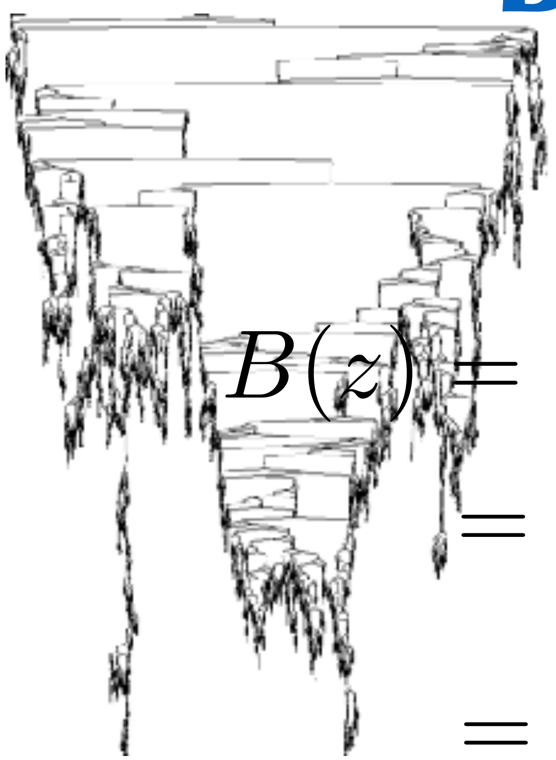
1. Numerical resolution with sufficient precision + algebraic manipulations
2. Local expansion (easy).
3. Easy.

Demo

IV. 2. Algebraic Generating Functions

Unambiguous Context-Free Languages

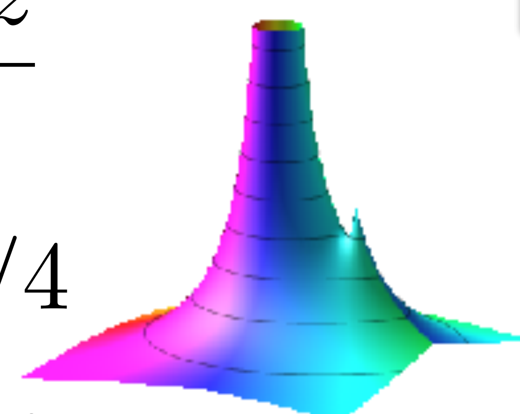
Basic Example: Binary Trees



$$\begin{aligned} B(z) &= 1 + zB(z)^2 \\ &= 1 + z + 2z^2 + 5z^3 + \cdots \\ &= \frac{1-\sqrt{1-4z}}{2z} \end{aligned}$$

A 3-Step Method:

1. Locate dominant singularities;
2. Compute local behaviour;
3. Translate into asymptotics.



1. sing. en $z = 1/4$
2. local behaviour:

$$B(z) \underset{z\to 1/4}{=} 2 - 2(1-4z)^{1/2} + \cdots$$

3. translation:

$$B_n \sim \frac{4^n n^{-3/2}}{\sqrt{\pi}} \left(1 - \frac{9}{8n} + \cdots\right)$$

Probability that a leaf is a child of the root?

$$\frac{[z^n]2zB(z)}{B_n} = \frac{1}{2} + \frac{3}{4n} + O\left(\frac{1}{n^2}\right)$$

Demo.

Algebraic Generating Functions

$$P(z, y(z)) = 0$$

1a. Location of possible singularities
Implicit Function Theorem:

$$P(z, y(z)) = \frac{\partial P}{\partial y}(z, y(z)) = 0 \quad \text{(discriminant)}$$

Numerical resolution
with sufficient precision
+ algebraic manipulations

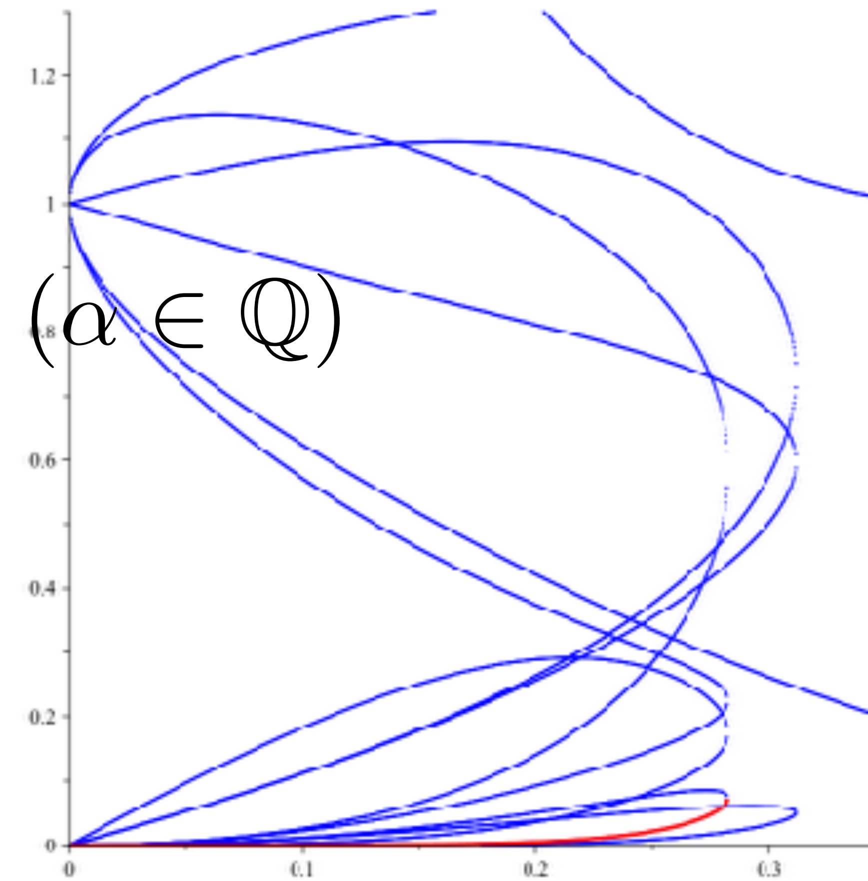
1b. Analytic continuation
finds the dominant ones

2. Local behaviour (Puiseux): $(1 - z/\rho)^{\alpha}$, $(\alpha \in \mathbb{Q})$

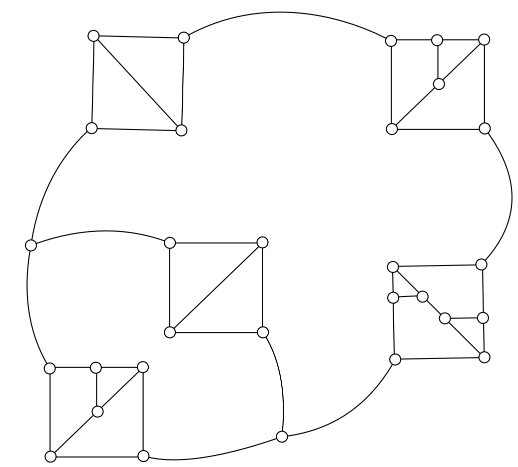
3. Translation: easy:

$$a_n \underset{n\to\infty}{\sim} c\,\rho^{-n} \frac{n^{-\alpha-1}}{\Gamma(-\alpha)}$$

with c, ρ algebraic, α rational.



3-regular 2-connected Planar Graphs



$$U = 2G_3 + T + 2U^2 = \frac{T}{(1-U)^3}, T = z(1+B)^3, B = \frac{G_3 + B^2}{1+B} + z\left(B + \frac{1}{2}B^2\right)$$

define power series $U(z), G_3(z), T(z), B(z)$.

The aim is to compute the asymptotic behaviour of $[z^n]B(z)$.

1. Eliminating U,T,G_3 gives $P = 16B^6z^2 + \cdots + z^2(z^2 + 11z - 1)$.
2. The discriminant has degree 20, but *only one root in (0,1]*:

   $\rho \approx .102$ root of $54z^3 + 324z^2 - 4265z + 432$.
3. At $z = \rho$, P has *only 1 (double) real positive* root: $B(\rho)$
4. Computing more terms gives

   $B(z) = B(\rho) + c_1\left(1 - \frac{z}{\rho}\right) \pm c\left(1 - \frac{z}{\rho}\right)^{3/2} + \cdots$ with an explicit c
5. Conclusion:

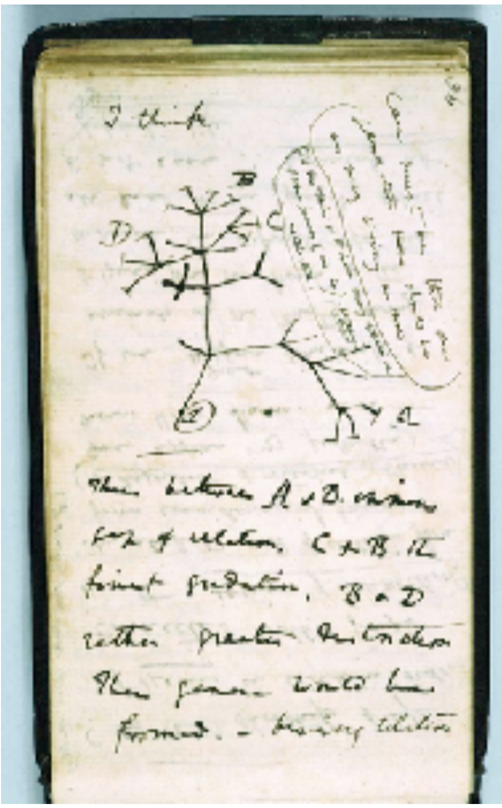
   $$[z^n]B(z) \sim \frac{3c}{4\sqrt{\pi}} n^{-5/2} \rho^{-n}.$$

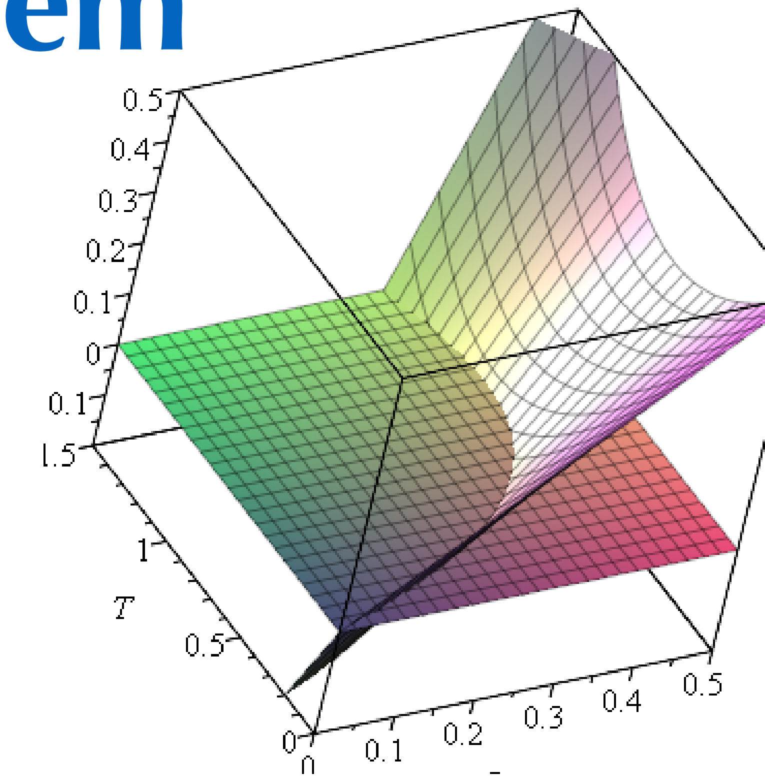
Analytic continuation exploiting the combinatorial origin.

IV. 3. More General Implicit Generating Functions

Often behave like algebraic ones

Implicit Function Theorem Locates Singularities





$$\mathcal{T} = \mathcal{Z} \times \mathrm{SET}(\mathcal{T})$$

$$T(z) = z \exp(T(z))$$

Singularities may only occur when $\partial/\partial T = 0$

$$1 = z\exp(T) \quad \Rightarrow T = 1, z = e^{-1}$$

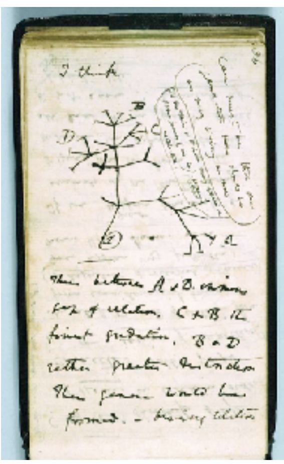
For a system $Y = H(z, Y)$ the condition becomes

$$\det\left(\mathrm{Id} - \frac{\partial H}{\partial Y}\right) = 0$$

Jacobian matrix

Algebraic-Like Singular Behaviour: Drmota-Lalley-Woods Theorem

$$T(z) = z\exp(T(z))$$



$$z = T\exp(-T) \qquad \textit{expansion at } T{=}1$$

$$= e^{-1} - e^{-1}\frac{(T-1)^2}{2} + e^{-1}\frac{(T-1)^3}{3} + \cdots$$

no linear term by the IFT

$$T = 1 - \sqrt{2}\boxed{\sqrt{1-ez}} + O(1-ez).$$

Singularity analysis → $\dfrac{T_n}{n!} \sim \dfrac{e^n}{\sqrt{2\pi}n^{3/2}}$ (aka Stirling's formula)

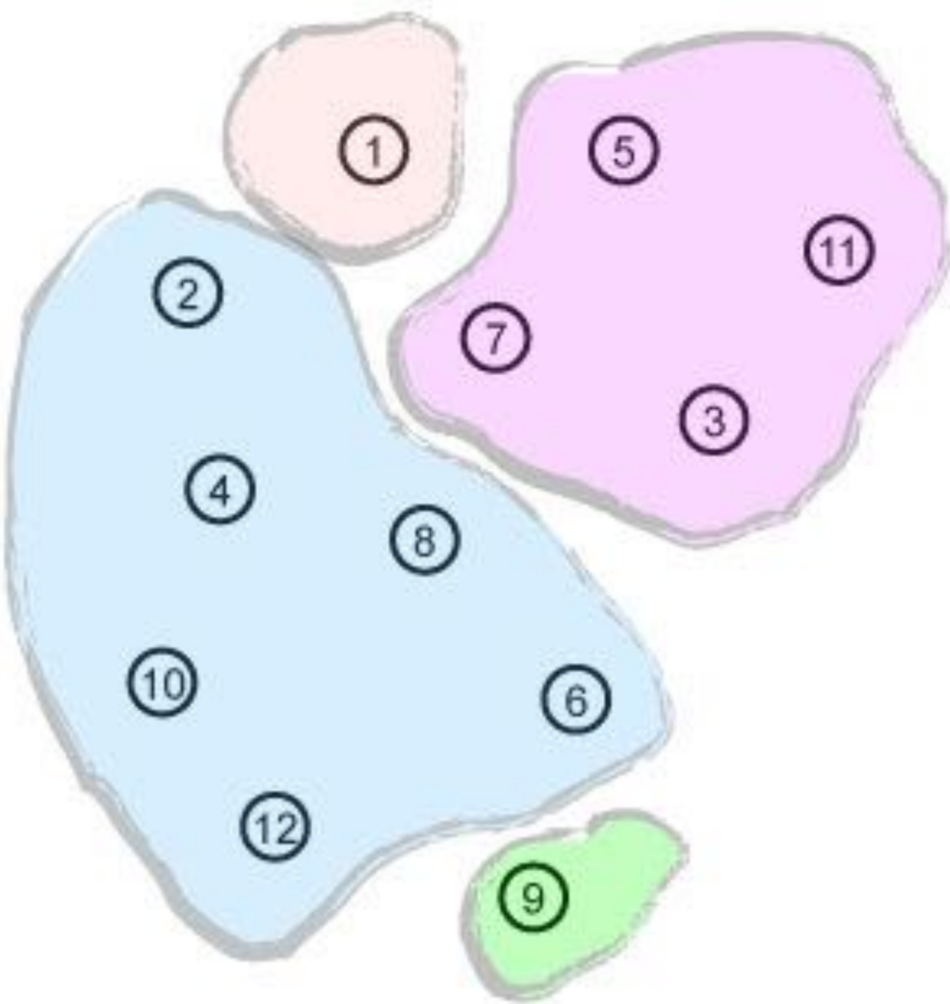
General Case [DLW] Under mild and common conditions, the singularities of combinatorial systems behave like square roots.

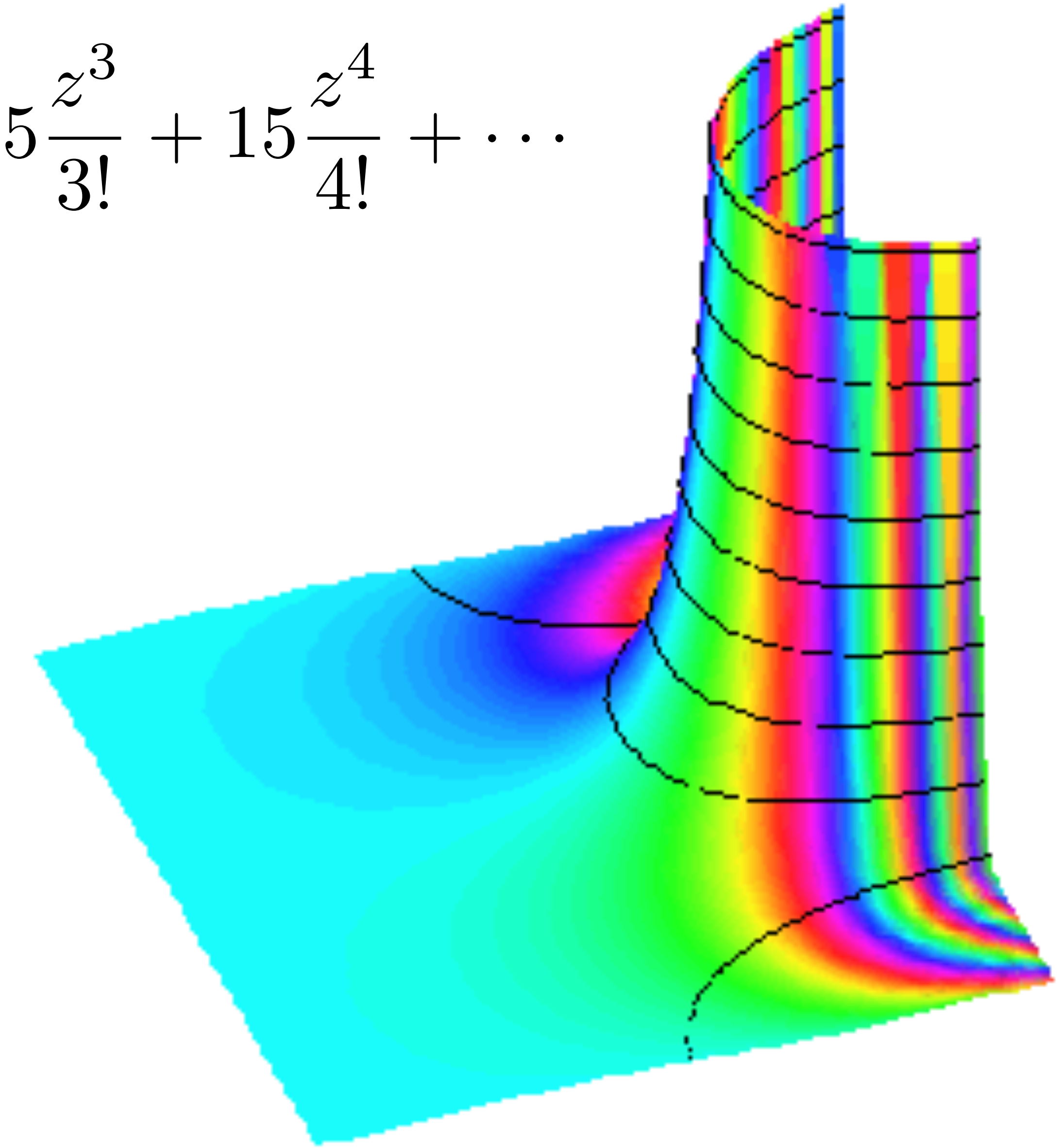
IV. 4. Faster Growth & Saddle-Point Method

Example: Set Partitions

$$\text{Part} = \text{Set}(\text{Set}_{>0}(\mathcal{Z}))$$

$$\exp(e^z - 1) = 1 + 1\frac{z}{1!} + 2\frac{z^2}{2!} + 5\frac{z^3}{3!} + 15\frac{z^4}{4!} + \cdots$$

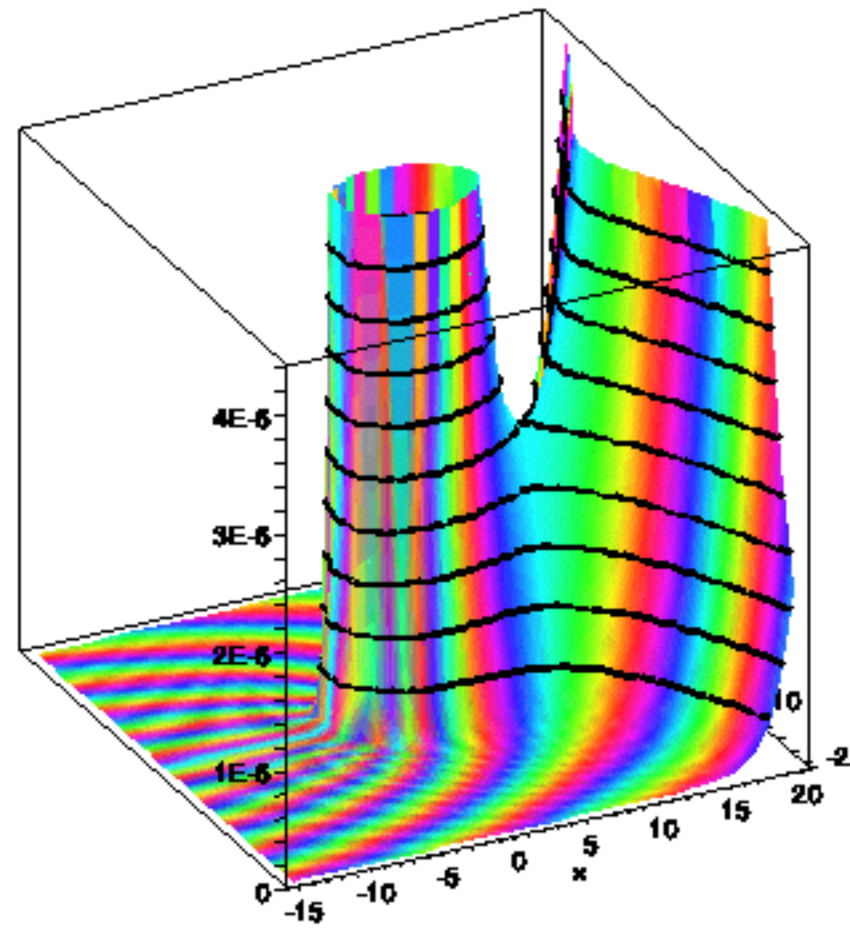




Saddle-Point Method

(Functions with fast singular growth)

$$a_n = \frac{1}{2\pi i} \oint \underbrace{\frac{f(z)}{z^{n+1}}}_{\exp(h(z))} dz$$



1. Saddle-point equation $h'(R_n) = 0$

2. Change of variables $h(z) = h(R_n) - u^2$

3. Termwise integration

$$f_n \approx \frac{f(R_n)}{R_n^{n+1}\sqrt{2\pi h''(R_n)}}$$

4. Sufficient conditions: next slide

Ex: f=exp

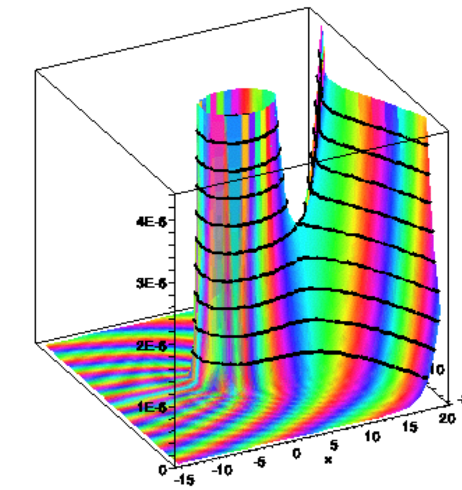
$$h(z) = z - (n+1)\log z$$

$$R_n = n+1$$

$$\frac{1}{n!} \approx \frac{e^{n+1}}{(n+1)^{n+1}\sqrt{\frac{2\pi}{n+1}}}$$

Stirling!

Hayman Admissibility

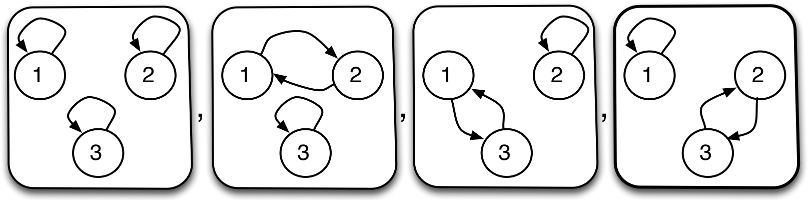


A set of easy-to-use sufficient conditions

f admissible $\Longrightarrow$ the formula of the previous slide holds

> **Thm.** f,g admissible, P polynomial. Then,
> 1. exp(f), fg and f+P admissible;
> 2. lc(P)>0 $\Rightarrow$ fP and P(f) admissible;
> 3. if e^P has ultimately >0 coeffs, then it is admissible.

Exs: sets (e^z), involutions ($e^{z+z^2/2}$), set partitions ($\exp(e^z - 1)$).

$\mathrm{Inv}[\{1,2,3\}] = \Big\{$  $\Big\}$

[Hayman 1956; see also Wyman 1959]

Conclusions

If you can specify it, you can analyze it!

Permutations	Languages
Mappings	Integers
Words	Compositions
Strings	Partitions
Urns	...
Trees	

+ *parameters*

Limit Laws

Universality phenomena:

Ex.: # trees of various types $\longrightarrow K\rho^n n^{-3/2} \longrightarrow$ pathlength in $n^{1/2}$

Summary & Conclusion

What we have:

Automated?

Combinatorial specification →
- counting; ✓
- random generation; ✓
- asymptotic analysis. ✗ (not fully)

Ultimate (dream) goal

The End

Combinatorial specification + Algorithm → Automatic complexity analysis

Where to learn more:

On-line course by R. Sedgewick at
http://ac.cs.princeton.edu/online/