From Analysis of Algorithms to Analytic Combinatorics

a journey with Philippe Flajolet

Robert Sedgewick
Princeton University

This talk is dedicated to the memory of Philippe Flajolet

Philippe Flajolet 1948–2011

From *Analysis of Algorithms* to *Analytic Combinatorics*

47. PF. Elements of a general theory of combinatorial structures. 1985.

69. PF. Mathematical methods in the analysis of algorithms and data structures. 1988.

70. PF L'analyse d'algorithmes ou le risque calculé. 1986.

72. PF. Random tree models in the analysis of algorithms. 1988.

88. PF and Michèle Soria. Gaussian limiting distributions for the number of components in combinatorial structures. 1990.

91. Jeffrey Scott Vitter and PF. Average-case analysis of algorithms and data structures. 1990.

95. PF and Michèle Soria. The cycle construction.1991.

97. PF. Analytic analysis of algorithms. 1992.

99. PF. Introduction à l'analyse d'algorithmes. 1992.

112. PF and Michèle Soria. General combinatorial schemas: Gaussian limit distributions and exponential tails. 1993.

130. RS and PF. An Introduction to the Analysis of Algorithms. 1996.

131. RS and PF. Introduction à l'analyse des algorithmes. 1996.

171. PF. Singular combinatorics. 2002.

175. Philippe Chassaing and PF. Hachage, arbres, chemins & graphes. 2003.

189. PF, Éric Fusy, Xavier Gourdon, Daniel Panario, and Nicolas Pouyanne. A hybrid of Darboux's method and singularity analysis in combinatorial asymptotics. 2006.

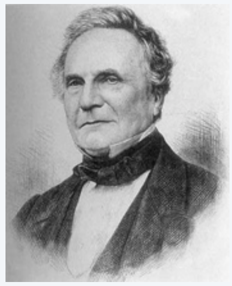
192. PF. Analytic combinatorics—a calculus of discrete structures. 2007.

201. PF and RS. Analytic Combinatorics. 2009.

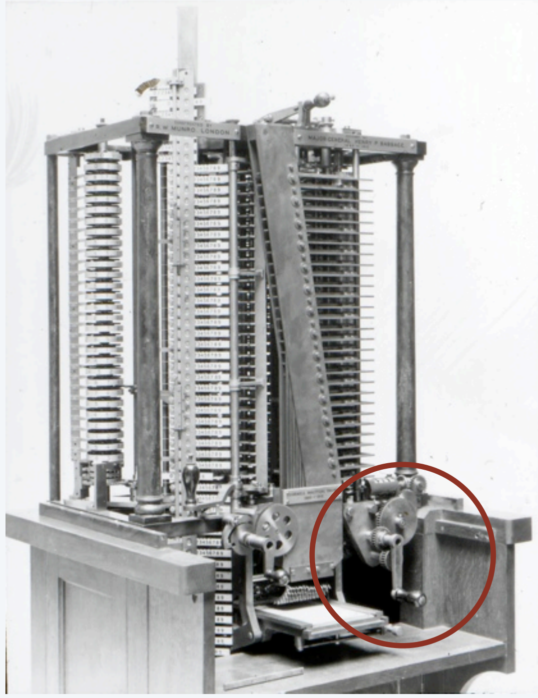
Analysis of Algorithms

Pioneering research by Knuth put the study of the performance of computer programs on a scientific basis.

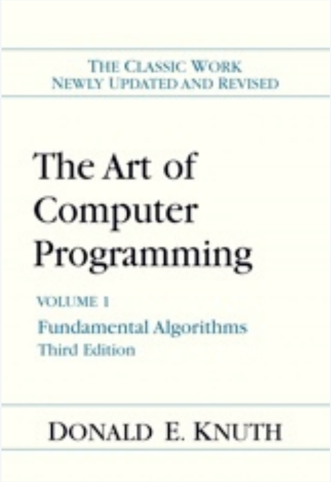
"As soon as an Analytic Engine exists, it will necessarily guide the future course of the science. Whenever any result is sought by its aid, the question will arise - By what course of calculation can these results be arrived at by the machine in the shortest time?"

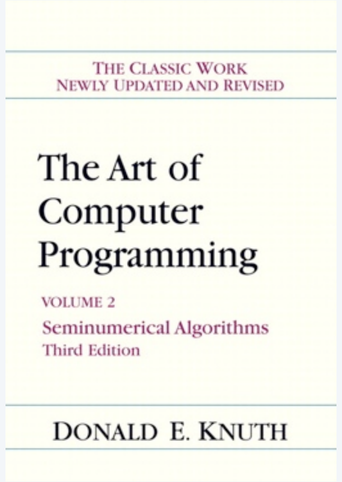


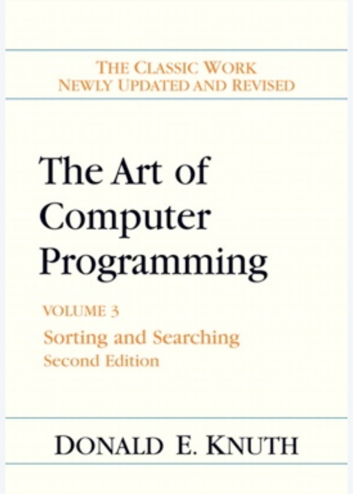
Charles Babbage



how many times do I have to turn the crank?







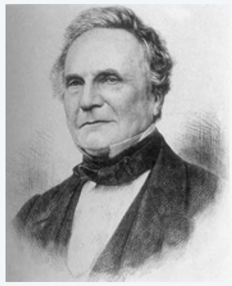
Challenge: Keep pace with explosive growth of new algorithms

a full employment theorem for algorithm analysts

Analysis of Algorithms

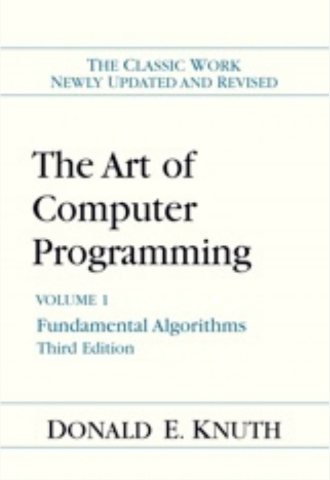
Pioneering research by Knuth put the study of the performance of computer programs on a scientific basis.

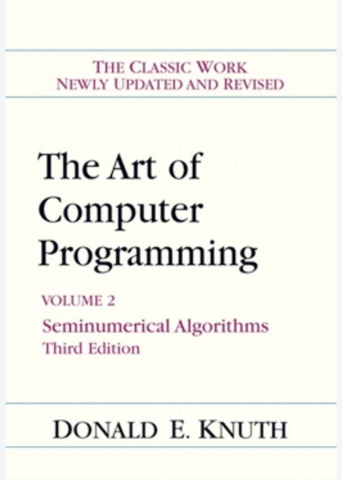
> *"As soon as an Analytic Engine exists, it will necessarily guide the future course of the science. Whenever any result is sought by its aid, the question will arise - By what course of calculation can these results be arrived at by the machine in the shortest time?"*

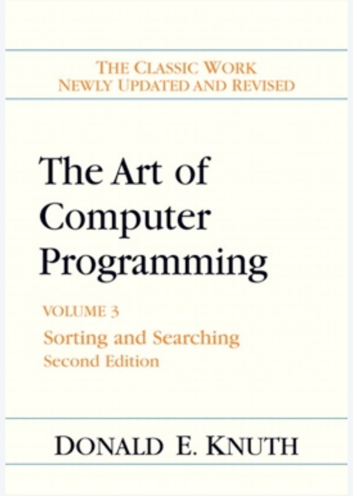


Charles Babbage

how long will my cellphone's battery last?







Challenge: Keep pace with explosive growth of new algorithms

a full employment theorem for algorithm analysts ***AND STILL VALID***

Genesis of “Analytic Combinatorics” (early 1980s)

Optimism and opportunity

Knuth volumes 1-3

Search for generality

Algorithms for the masses

Teaching and research in AofA

TeX

$$\frac{2}{\ln 2}\Gamma(\frac{k\pi i}{\ln 2})\zeta(\frac{2k\pi i}{\ln 2},\frac{1}{4})$$

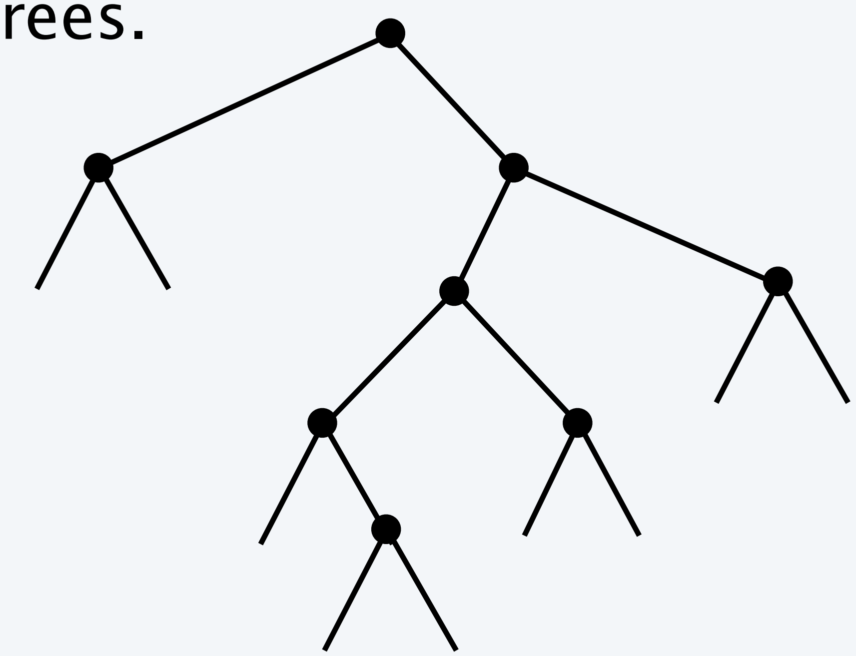
Main idea:

Teach the basics so CS students can get started on AofA.

Analysis of algorithms: classic example I

A binary tree is a node connected to two binary trees.

How many binary trees with N nodes?



Develop a recurrence relation.

$$B_N = \sum_{0 \le k < N} B_k B_{N-1-k} \qquad B_0 = 0$$

Then introduce a generating function.

$$B(z) = \sum_{k \ge 0} z^k$$

Multiply both sides by z^N and sum to get an equation

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

that we can solve algebraically

$$B(z) = \frac{1 + \sqrt{1 - 4z}}{2z}$$

Quadratic equation

and expand to get coefficients

$$B_N = \frac{1}{N+1}\binom{2N}{N}$$

Binomial theorem

that we can approximate.

$$B_N \sim \frac{4^N}{N\sqrt{\pi N}}$$

Stirling's approximation

Challenge: Efficiently teach basic math skills behind such derivations.

Analysis of algorithms: classic example II

A binary search tree is a binary tree with keys in order in inorder.

Path length of a BST built from N random distinct keys?

Develop a recurrence relation.

$$C_N = N - 1 + \frac{1}{N} \sum_{1 \le k \le N} (C_k + C_{N-k-1}) \qquad C_0 = 0$$

Then introduce a generating function.

$$C(z) = \sum_{k \ge 0} z^k$$

Multiply both sides by z^N and sum to get an equation

$$C'(z) = \frac{1}{(1-z)^2} + \frac{2}{1-z} C(z)$$

that we can solve

$$C(z) = \frac{2}{(1-z)^2} \ln \frac{1}{1-z}$$

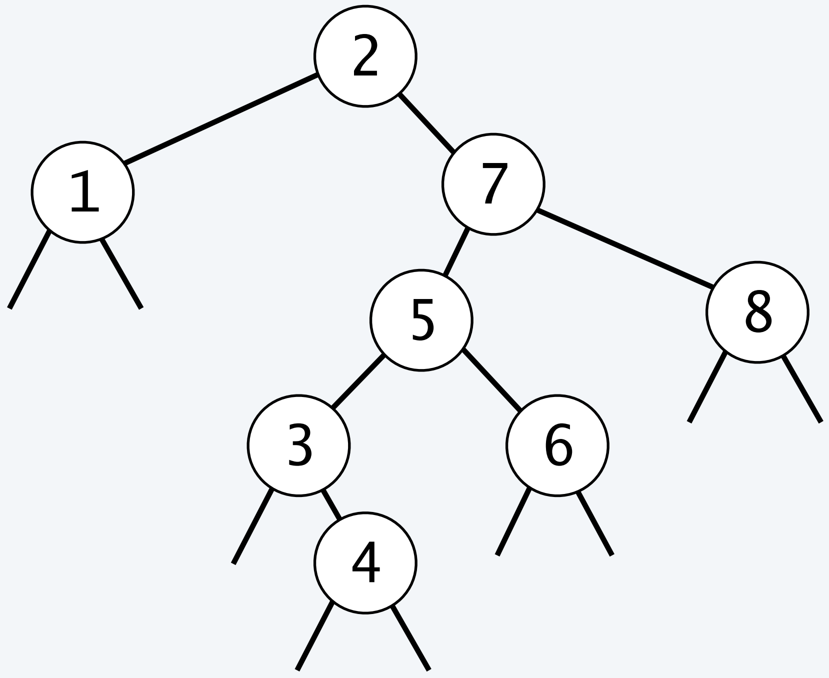
and expand to get coefficients

$$C_N = 2(N+1)(H_{N+1} - 1)$$

that we can approximate.

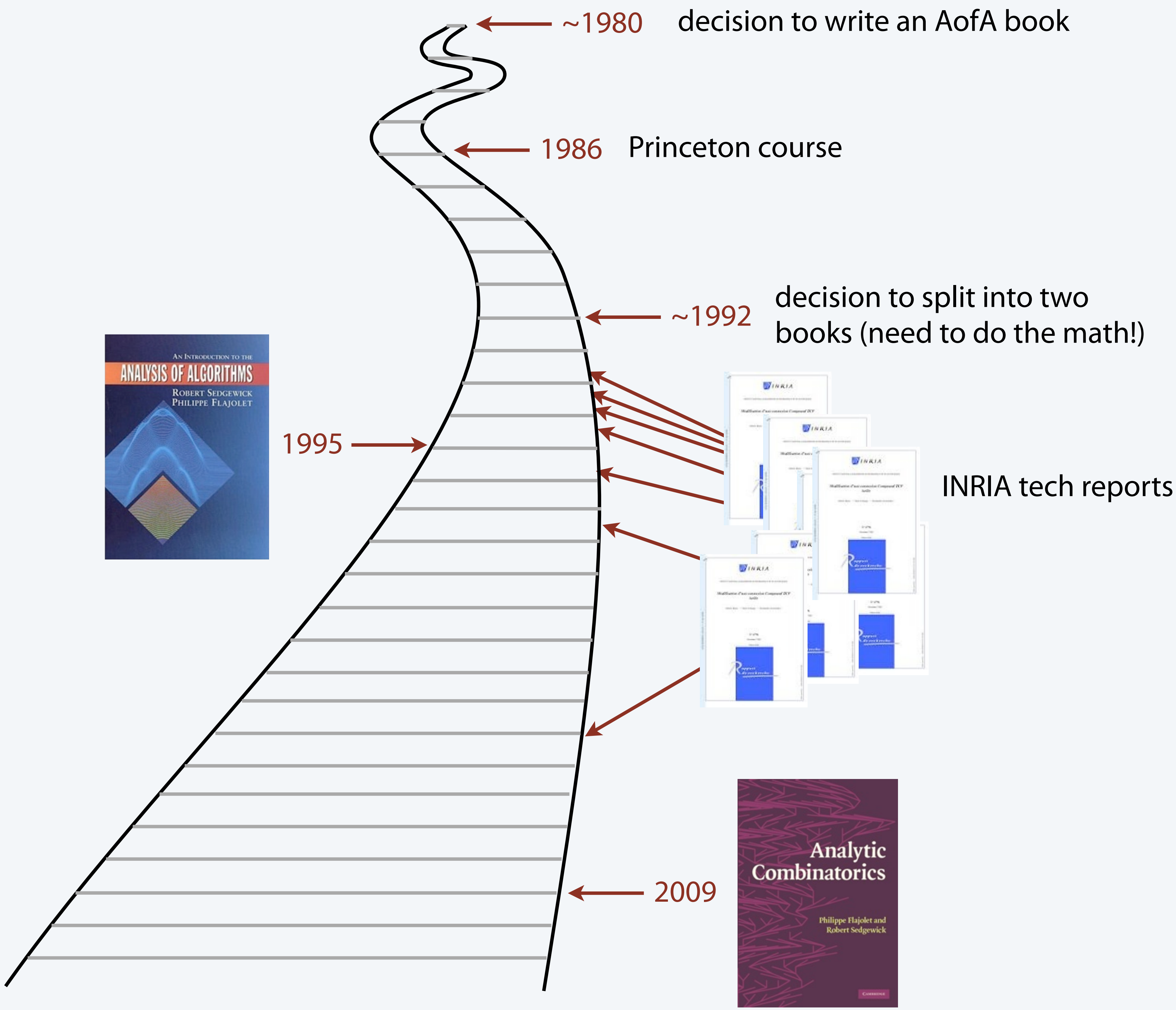
$$C_N \sim 2N \ln N$$

Euler-MacLaurin summation



Note: Analyzing a property of permutations, not counting trees.

Thirty years in the making



Analysis of Algorithms

Goal: Teach the mathematical concepts needed to study the performance of computer programs.

Recurrences

1st order, nonlinear, higher order, divide-and conquer

Generating Functions

OGFs, EGFs, solving recurrences, CGFs, symbolic method, Lagrange inversion, PGFs, BGFs, special functions.

Asymptotics

expansions, Euler-Maclaurin summation, bivariate, Laplace method, normal approximations, Poisson approximations, GF asymptotics

Trees

forests, BSTs, Catalan trees, path length, height, unordered, labelled, 2-3

Permutations

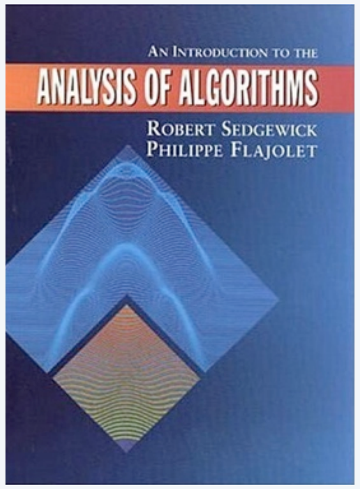
properties, representations, enumerations, inversions, cycles, extremal parameters

Strings and Tries

bitstrings, REs, FSAs, KMP algorithm, context-free grammars, tries.

Words and Maps

hashing, birthday paradox, coupon collector, occupancy, maps, applications



Teaches the basics for CS students to get started on AofA. ✓

Done?

An emerging idea (PF, 1980s)

In principle, classical methods can provide

- full details
- full and accurate asymptotic estimates

In practice, it is often possible to

- generalize specialized derivations
- skip details and move directly to accurate asymptotics

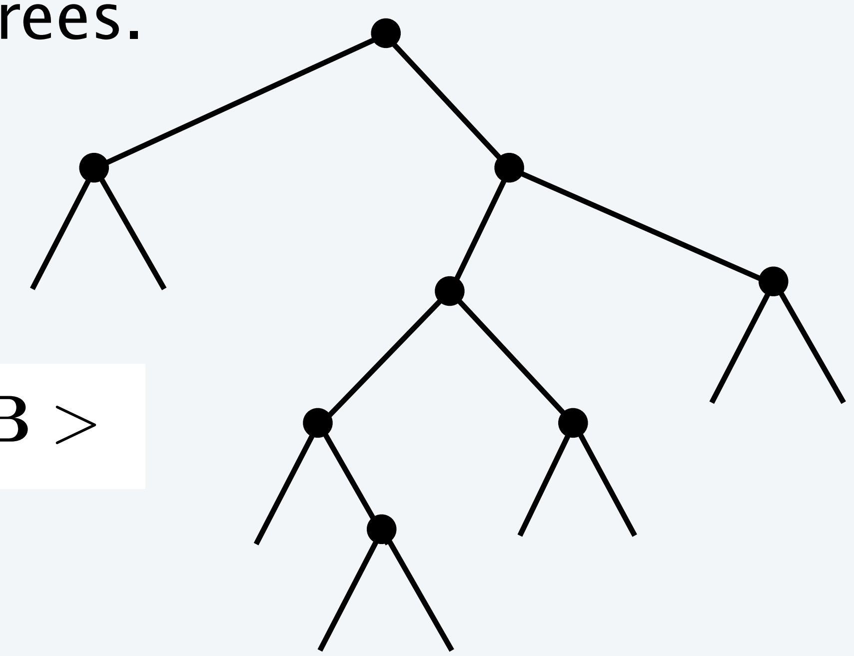
Ultimate (unattainable) goal: Automatic analysis of algorithms

Ex.

Analytic Combinatorics: classic example

A binary tree is a node connected to two binary trees.

How many binary trees with N nodes?



Develop a **combinatorial construction,**

$$<\mathbf{B}> = \epsilon + <\mathbf{B}> \times \bullet \times <\mathbf{B}>$$

which *directly maps* to a **GF equation**

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

that we can manipulate algebraically

$$B(z) = \frac{1+\sqrt{1-4z}}{2z}$$

and treat as a function in the complex plane *directly approximate* via **singularity analysis**

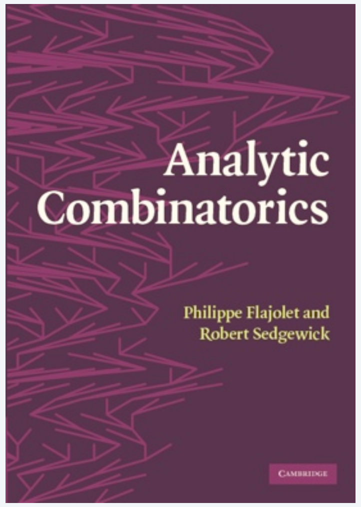
$$B_N = \frac{4^N}{N\Gamma(1/2)\sqrt{N}} \sim \frac{4^N}{N\sqrt{\pi N}}$$

Challenge: Develop an effective calculus for such derivations.

Note: Construction for BSTs is not so simple.

Analytic Combinatorics

A calculus of discrete structures



Symbolic Methods

- Combinatorial Structures and OGFS
- Labelled Structures and EGFs
- Parameters and MGFs

Complex Asymptotics

- Rational and Meromorphic Asymptotics
- Singularity Analysis
- Saddle-Point Asymptotics

Random Structures

- Multivariate Asymptotics
- Limit Laws

The Symbolic Method

Constructions of combinatorial classes can be **automatically translated** to GF definitions.

Unlabelled classes lead to **O**GFs

Ex. Cartesian product

Construction for **ordered pairs,**

$$< \mathbf{A} > = < \mathbf{B} > \times < \mathbf{C} >$$

class of objects $\alpha = (\beta, \gamma)$ *where* $\beta \in < \mathbf{B} >$ *and* $\gamma \in < \mathbf{C} >$

a corresponding **counting sequence,**

$$A_n$$

and an **ordinary generating function (OGF).**

$$A(z) = \sum_{k \geq 0} A_k z^k = \sum_{\alpha \in \mathbf{A}} z^{|\alpha|}$$

OGF of Cartesian product is product of OGFs by distributive law

$$A(z) = \sum_{(\beta,\gamma) \in \mathbf{B} \times \mathbf{C}} z^{|\beta|+|\gamma|} = \sum_{\beta \in \mathbf{B}} z^{|\beta|} \sum_{\gamma \in \mathbf{C}} z^{|\gamma|} = B(z) \cdot C(z)$$

The Symbolic method

Constructions of combinatorial classes can be automatically translated to GF definitions.

Unlabelled classes lead to OGFs

	Construction	*OGF*
Union	$<\mathbf{A}>=<\mathbf{B}>+<\mathbf{C}>$	$A(z)=B(z)+C(z)$
Product	$<\mathbf{A}>=<\mathbf{B}>\times<\mathbf{C}>$	$A(z)=B(z)\cdot C(z)$
Sequence	$<\mathbf{A}>=\mathbf{SEQ}(<\mathbf{B}>)$	$A(z)=\dfrac{1}{1-B(z)}$
Powerset	$<\mathbf{A}>=\mathbf{PSET}(<\mathbf{B}>)$	$A(z)=\exp(B(z)-B(z)^2/2+B(z)^3/3+\ldots$
Multiset	$<\mathbf{A}>=\mathbf{MSET}(<\mathbf{B}>)$	$A(z)=\exp(B(z)+B(z)^2/2+B(z)^3/3+\ldots$
Cycle	$<\mathbf{A}>=\mathbf{CYC}(<\mathbf{B}>)$	$A(z)=\ln\dfrac{1}{1-B(z)}+\dfrac{1}{2}\ln\dfrac{1}{1-B(z)^2}+\ldots$
[several others]		

The Symbolic Method

Elementary OGF examples (unlabelled objects).

Nonnegative Integers

an integer (in unary)

1 1 1 1 1 1 1 1 1 1 1 1 1 1

atomic class

$$< \mathbf{Z} >:= 1$$

construction for class of all nonnegative integers

$$< \mathbf{I}_{\geq 0} >= \mathbf{SEQ}(< \mathbf{Z} >)$$

automatic derivation of OGF

$$Z(z) = z$$

$$I_{\geq 0}(z) = \frac{1}{1-z}$$

$$\frac{1}{1-z} = 1 + z + z^2 + z^3 \ldots = \sum_{N \geq 0} z^N$$

✓ *one object of each size*

Binary words

a binary word

0 1 1 0 1 1 0 1 1 1 1 1 1

atomic class

$$< \mathbf{Z} >= 0 \mid 1$$

construction for class of all binary words

$$< \mathbf{W} >= \mathbf{SEQ}(< \mathbf{Z} > + < \mathbf{Z} >)$$

automatic derivation of OGF

$$Z(z) = z$$

$$W(z) = \frac{1}{1-(Z(z)+Z(z))} = \frac{1}{1-2z}$$

$$\frac{1}{1-2z} = 1 + 2z + 4z^2 + 8z^3 \ldots = \sum_{N \geq 0} 2^N z^N$$

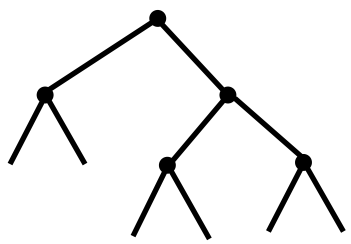
✓ *2^N objects of each size*

The Symbolic Method

Representative OGF examples (unlabelled objects).

Binary trees

a binary tree



atomic class

$$<\mathbf{Z}>:=\bullet$$

construction for class of all binary trees

$$<\mathbf{B}>=\epsilon+<\mathbf{B}>\times<\mathbf{Z}>\times<\mathbf{B}>$$

automatic derivation of OGF

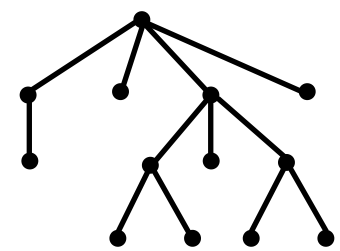
$$B(z)=1+zB(z)^2$$

$$zB(z)^2-B(z)+1=0$$

$$B(z)=\frac{1+\sqrt{1-4z}}{2z}$$

Trees

a general tree



atomic class

$$<\mathbf{Z}>:=\bullet$$

construction for class of all ge

$$<\mathbf{G}>=<\mathbf{Z}>\mathbf{SEQ}(<\mathbf{G}>)$$

automatic derivation of OGF

$$G(z)=\frac{z}{1-G(z)}$$

$$G(z)^2-G(z)+z=0$$

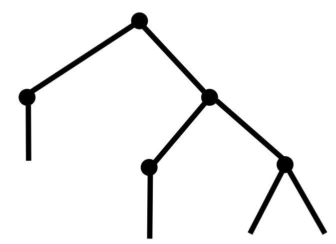
$$G(z)=\frac{1+\sqrt{1-4z}}{2}$$

The Symbolic Method

Derivations are easy to generalize.

Unary-Binary Trees

a unary-binary tree



atomic class

$$< \mathbf{Z} > := \bullet$$

construction for class of all unary-binary trees

$$< \mathbf{U} > = < \mathbf{Z} > (< 1 > + < \mathbf{U} > + < \mathbf{U} > \times < \mathbf{U} >)$$

automatic derivation of OGF

$$U(z) = z(1 + U(z) + U(z)^2)$$

$$U(z) = \frac{1 - z - \sqrt{(1+z)(1-3z)}}{2z}$$

The Symbolic method

Constructions of combinatorial classes can be automatically translated to GF definitions.

Labelled classes lead to EGFs

		Construction	*EGF*
Union		$<\mathbf{A}>=<\mathbf{B}>+<\mathbf{C}>$	$A(z)=B(z)+C(z)$
Product		$<\mathbf{A}>=<\mathbf{B}>\star<\mathbf{C}>$	$A(z)=B(z)\cdot C(z)$
Sequence	*size k*	$<\mathbf{A}>=\mathbf{SEQ_k}(<\mathbf{B}>)$	$A(z)=B(z)^k$
	any size	$<\mathbf{A}>=\mathbf{SEQ}(<\mathbf{B}>)$	$A(z)=\dfrac{1}{1-B(z)}$
Set	*size k*	$<\mathbf{A}>=\mathbf{SET_k}(<\mathbf{B}>)$	$A(z)=B(z)^k/k!$
	any size	$<\mathbf{A}>=\mathbf{SET}(<\mathbf{B}>)$	$A(z)=e^{B(z)}$
Cycle	*size k*	$<\mathbf{A}>=\mathbf{CYC_k}(<\mathbf{B}>)$	$A(z)=B(z)^k/k$
	any size	$<\mathbf{A}>=\mathbf{CYC}(<\mathbf{B}>)$	$A(z)=\ln\dfrac{1}{1-B(z)}$

[several others]

The Symbolic Method

Representative EGF examples (labelled objects).

Derangements

a derangement

1	2	3	4	5	6	7	8
3	4	8	2	6	7	5	1

alternate (cycle) representation

{1 3 8} {2 4} {5 6 7}

construction for class of all derangements

$$<\mathbf{D}>=\mathbf{SET}(\mathbf{CYC}_{>1}(<\mathbf{Z}>))$$

automatic derivation of EGF

$$D(z)=e^{\sum_{k>1} z^k/k}=e^{\ln\frac{1}{1-z}-z}$$

$$=\frac{e^{-z}}{1-z}$$

Surjections

a surjection (onto mapping)

1	2	3	4	5	6	7	8
2	1	2	3	5	3	4	3

alternate (preimage) representation

1	2	3	4	5
{2}	{1, 3}	{4, 6, 8}	{7}	{5}

construction for class of all surjections

onto any initial segment of the integers

$$<\mathbf{R}>=\mathbf{SEQ}(\mathbf{SET}_{\geq 1}(<\mathbf{Z}>))$$

automatic derivation of EGF

$$R(z)=\frac{1}{1-\sum_{k\geq 1} z^k/k!}=\frac{1}{1-(e^z-1)}$$

$$=\frac{1}{2-e^z}$$

Recovering coefficients from GFs

is sometimes easy, but often challenging

examples (increasing order of difficulty)

class	GF	expansion
binary words	$W(z) = \dfrac{1}{1-2z}$	$1 + 2z + (2z)^2 + (2z)^3 + \ldots$
trees	$G(z) = \dfrac{1+\sqrt{1-4z}}{2}$	*binomial*
BSTs	$C(z) = \dfrac{2}{(1-z)^2} \ln \dfrac{1}{1-z}$	*elementary convolution*
permutations with all cycle lengths > 3	$D^{(3)}(z) = \dfrac{e^{z+z^2/2+z^3/3}}{(1-z)}$	*triple convolution*
unary-binary trees	$U(z) = \dfrac{1-z-\sqrt{(1+z)(1-3z)}}{2z}$	*not elementary*

Complexification

Assigning complex values to the variable z in a GF gives a method of analysis to estimate the coefficients.

The singularities of the function determine the method.

singularity type	*method of analysis*
meromorphic (just poles)	Cauchy (elementary)
fractional powers logarithmic	Cauchy (Flajolet-Odlyzko)
none (entire function)	saddle point

First Principle. Exponential growth of a function's coefficients is determined by the location of its singularities.

Second Principle. Subexponential factor in a function's coefficients is determined by the nature of its singularities.

Singularity Analysis

Flajolet-Odlyzko method provides detailed asymptotic estimates of coefficients for a broad function scale.

Ex. Fractional powers

Start with Cauchy coefficient formula

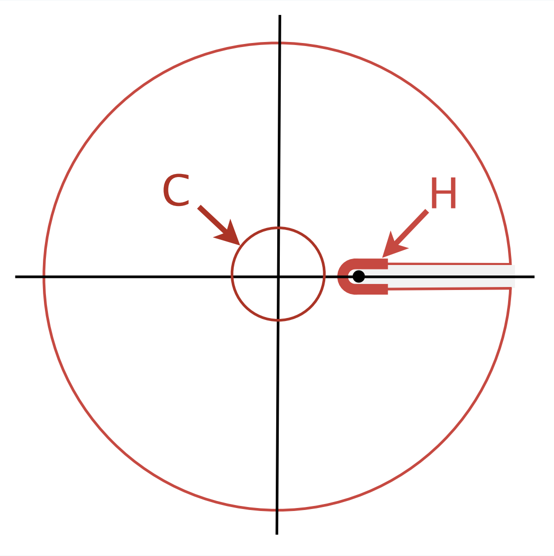
$$[z^N](1-z)^\alpha = \frac{1}{2\pi i}\int_C \frac{(1-z)^\alpha}{z^{N+1}}dz$$

deform to **Hankel countour**

$$\sim \frac{1}{2\pi i}\int_H \frac{(1-z)^\alpha}{z^{N+1}}dz$$

and evaluate, leading to integral representation of the Gamma function

$$\sim \frac{1}{\Gamma(\alpha)N^{\alpha+1}}$$



Approach extends to logarithmic factors.

Also effective for implicitly defined GFs.

Singularity Analysis

leads to general transfer theorems that *immediately provide* coefficient asymptotics.

$$[z^N]\frac{1}{(1-z/\rho)} = \rho^N$$

$$[z^N](1-z)^\alpha \sim \frac{1}{\Gamma(\alpha)N^{\alpha+1}}$$

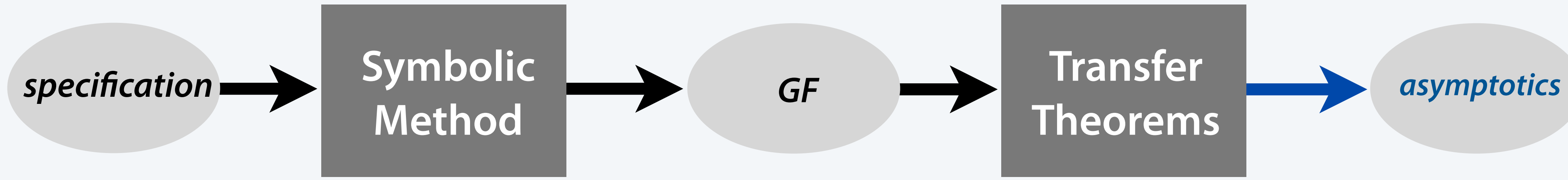
$$[z^N](1-z)^\alpha \ln\frac{1}{1-z} \sim \frac{1}{\Gamma(\alpha)N^{\alpha+1}}\ln N$$

Transfer theorems are effective even for approximations near singularities.

Complexification examples

class	GF	singularity type	at	coeffiecient asymptotics
binary words	$W(z) = \dfrac{1}{1-2z}$	*pole*	$\dfrac{1}{2}$	$W_N = 2^N$
derangements	$D(z) = \dfrac{e^{-z}}{1-z}$	*pole*	1	$D_N \sim e^{-1}$
surjections	$R(z) = \dfrac{1}{2-e^z}$	*poles*	$\ln 2$	$R_N \sim \dfrac{N!}{2(\ln 2)^{N+1}}$
trees	$G(z) = \dfrac{1+\sqrt{1-4z}}{2}$	*square root*	$\dfrac{1}{4}$	$G_N \sim \dfrac{4^{N-1}}{\sqrt{\pi N^3}}$
BSTs	$C(z) = \dfrac{2}{(1-z)^2} \ln \dfrac{1}{1-z}$	*logarithmic*	1	$C_N \sim 2N \ln N$

"If you can specify it, you can analyze it"



Representative examples

permutations with all cycle length > 3

$$< \mathbf{D}^{(3)} >= \mathbf{SET}(\mathbf{CYC}_{>3}(< \mathbf{Z} >)) \rightarrow D^{(3)}(z) = \frac{e^{z+z^2/2+z^3/3}}{(1-z)} \rightarrow D_N^{(3)} \sim e^{-1-1/2-1/3}$$

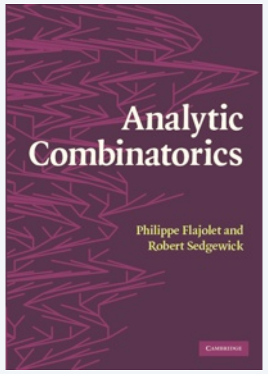
unary-binary trees

$$< \mathbf{U} >=< \mathbf{z} > (< 1 > + < \mathbf{U} > + < \mathbf{U} > \times < \mathbf{U} >)$$

$$\rightarrow U(z) = \frac{1 - z - \sqrt{(1+z)(1-3z)}}{2z} \rightarrow U_N \sim 3^N \sqrt{\frac{3}{4\pi N^3}}$$

AC Schemas

The symbolic method and singularity analysis admit universal laws of sweeping generality.



Ex. Context-free specifications

Develop a *system* of **combinatorial constructions,**

$$
\begin{aligned}
< \mathbf{G}_0 > &= OP_0(< \mathbf{G}_0 >, < \mathbf{G}_1 >, \ldots, < \mathbf{G_t} >)\\
< \mathbf{G}_1 > &= OP_1(< \mathbf{G}_0 >, < \mathbf{G}_1 >, \ldots, < \mathbf{G_t} >)\\
&\cdots\\
< \mathbf{G_t} > &= OP_t(< \mathbf{G}_0 >, < \mathbf{G}_1 >, \ldots, < \mathbf{G_t} >)
\end{aligned}
$$

Like a context-free language or data-type definition (irreducible and aperiodic)

which *directly maps* to a system of **GF equations**

$$
\begin{aligned}
G_0(z) &= F_0(G_0(z), G_1(z), \ldots, G_t(z))\\
G_1(z) &= F_1(G_0(z), G_1(z), \ldots, G_t(z))\\
&\cdots\\
G_t(z) &= F_t(G_0(z), G_1(z), \ldots, G_t(z))
\end{aligned}
$$

Symbolic method leads to a system of implicit function definitions

that we can manipulate algebraically

$$G_0(z) = F(G_0(z), G_1(z), \ldots, G_t(z))$$

Groebner basis elimination

to get a single complex function

$$G(z) \sim c - a\sqrt{1 - bz}$$

Drmota-Lalley-Woods theorem

that is amenable to **singularity analysis**

$$G_N \sim \frac{a}{2\sqrt{\pi N^3}} b^N$$

*A **universal** law for context-free specifications*

Bumps in the road

Constructions may be difficult to discover.
Ex: BSTs

Implicit functions may be difficult to analyze.
Ex: Counting balanced BSTs

Transfer theorems have *technical conditions* that need to be checked.
Ex: Planar graphs

Multiple dominant singularities lead to *oscillations*.
Ex: PF and RS "formula in common"

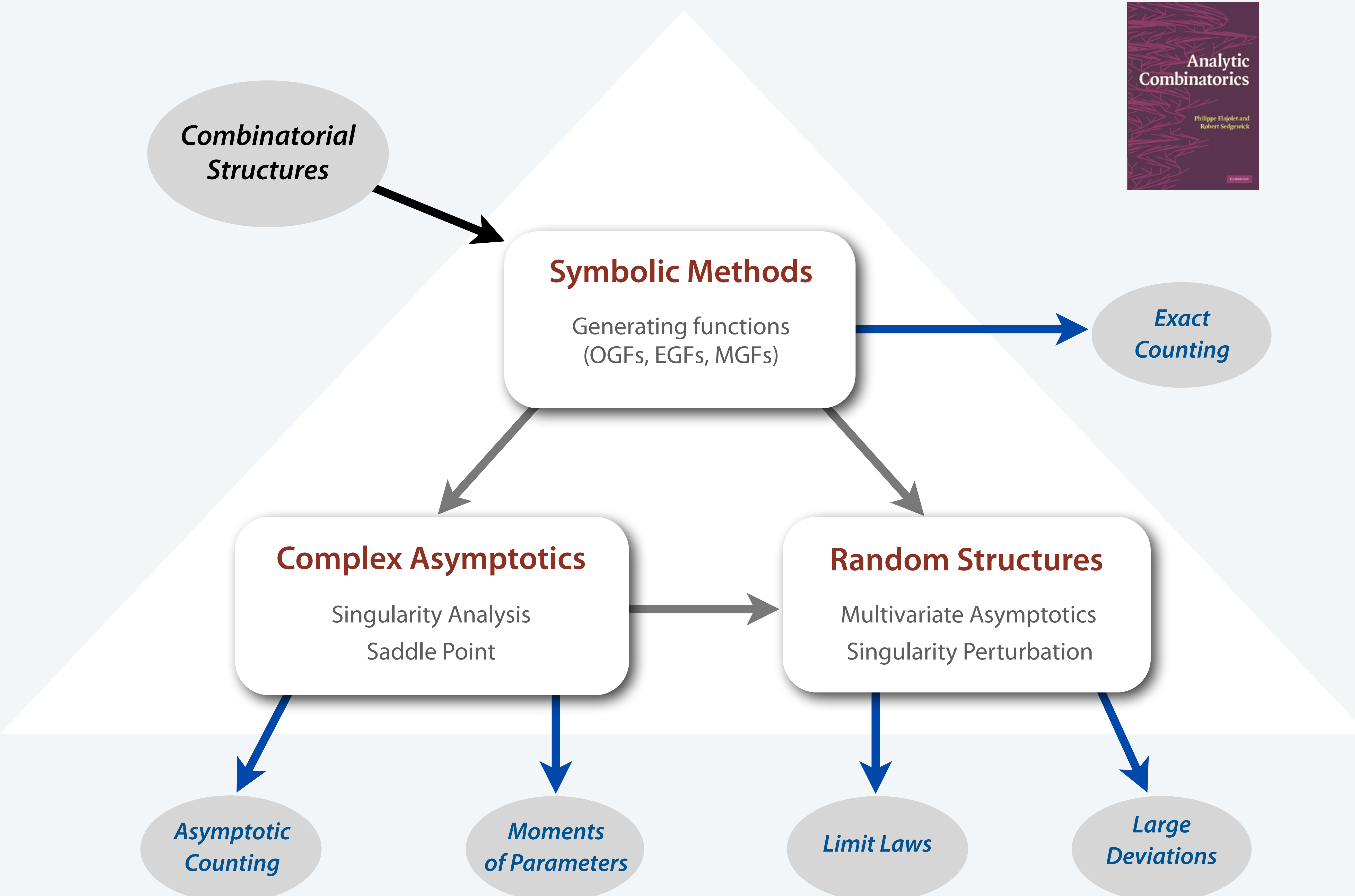
Many GFs have no singularities, need *saddle-point asymptotics.*
Ex: Involutions

Singularity structure may be complicated, need *Mellin asymptotics*.
Ex: Tries, Divide-and-conquer algorithms

AofA often requires studying properties, need *MGFs and limit laws.*
Ex: Arithmetic algorithms

Many of these have been effectively addressed and research is ongoing.

The Logical Structure of *Analytic Combinatorics*

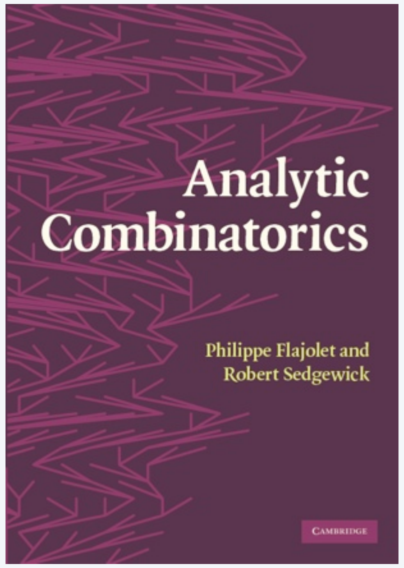


If you can specify it, you can analyze it

Applications of analytic combinatorics

- patterns in random strings
- polynomials over finite fields
- hashing
- digital tree and tries
- geometric search
- combinatorial chemistry
- arithmetic algorithms
- planar maps and graphs
- probabilistic stream algorithms
- master theorem for divide-and-conquer
- bioinformatics
- statistical physics

. . .



A calculus for the study of discrete structures. ✓

A brilliant mathematician and truly a computer scientist

PF, SODA 2007

Knuth proved the point that precise analysis is both feasible and fruitful, but his attention to detail was viewed as excessive by many.

Theoretical computer science reverted to worst-case analysis based on tools from computational complexity. In all too many cases, this has resulted in an excess of its own, with works culminating in teratological constrcutions both devoid of mathematical simplicity and elegance and bearing little relevance to the practice of computing.

At the same time, average-case and probabilistic analyses have proven to have spectacular impact on the practice of computing.

Many fundamental algorithms and data structures can be precisely analyzed and tuned for optimal performance. The corresponding calculus, largely motivated by considerations of algorithmic efficiency, is also of some mathematical interest per se.

From *Analysis of Algorithms* to *Analytic Combinatorics*

47. PF. Elements of a general theory of combinatorial structures. 1985.

69. PF. Mathematical methods in the analysis of algorithms and data structures. 1988.

70. PF L'analyse d'algorithmes ou le risque calculé. 1986.

72. PF. Random tree models in the analysis of algorithms. 1988.

88. PF and Michèle Soria. Gaussian limiting distributions for the number of components in combinatorial structures. 1990.

91. Jeffrey Scott Vitter and PF. Average-case analysis of algorithms and data structures. 1990.

95. PF and Michèle Soria. The cycle construction.1991.

97. PF. Analytic analysis of algorithms. 1992.

99. PF. Introduction à l'analyse d'algorithmes. 1992.

112. PF and Michèle Soria. General combinatorial schemas: Gaussian limit distributions and exponential tails. 1993.

130. RS and PF. An Introduction to the Analysis of Algorithms. 1996.

131. RS and PF. Introduction à l'analyse des algorithmes. 1996.

171. PF. Singular combinatorics. 2002.

175. Philippe Chassaing and PF. Hachage, arbres, chemins & graphes. 2003.

189. PF, Éric Fusy, Xavier Gourdon, Daniel Panario, and Nicolas Pouyanne. A hybrid of Darboux's method and singularity analysis in combinatorial asymptotics. 2006.

192. PF. Analytic combinatorics—a calculus of discrete structures. 2007.

201. PF and RS. Analytic Combinatorics. 2009.

Thank you, Philippe !

Working with you was a pleasure, an honor, and a privilege.

http://algo.inria.fr/pfac/PFAC/PFAC.html