"If You Can Specify It, You Can Analyze It"

the lasting legacy of Philippe Flajolet

Robert Sedgewick
Princeton University

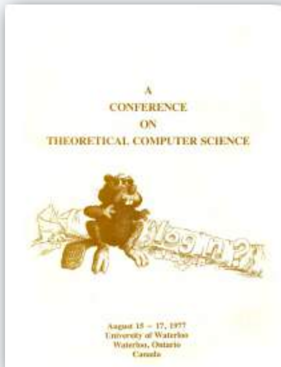
Dedicated to the memory of Philippe Flajolet

Philippe Flajolet 1948–2011

"If You Can Specify It, You Can Analyze It"

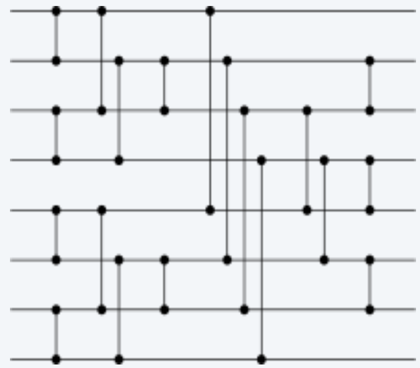
- **Brief History**
- Analysis of Algorithms
- Analytic Combinatorics
- Flajolet Collected Works
- New Directions

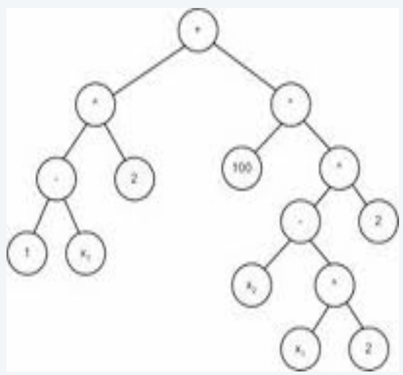
PF, 1977: “I believe that we have a formula in common!”



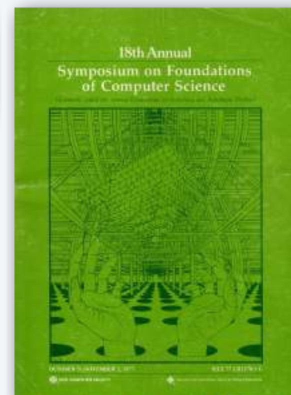
Data Movement in Odd-Even Merging
by Robert Sedgewick ....

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



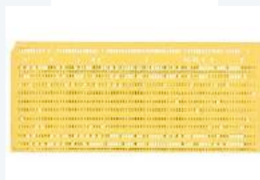
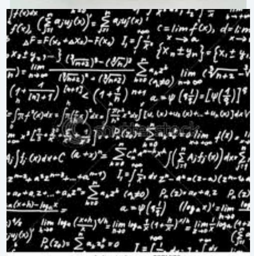


$$\frac{2k\pi i - \log 2}{\log 2}\,\Gamma\left(\frac{k\pi i}{\log 2}\right)\zeta\left(\frac{2\,k\pi i}{\log 2}\right)$$



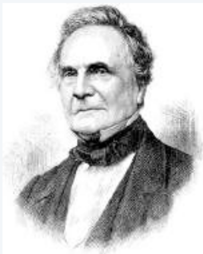
On the Average Number of Registers Required for Evaluating Arithmetic Expressions
by P. Flajolet, J. C. Raoult, and J. Vuillemin ....

Coming of age in CS (RS and PF generation)

when we entered school		when we started work	
transistors		integrated circuits	
punched cards		terminals	
typewriter		word processing	
Math		CS	

A more profound change than PCs or the internet.

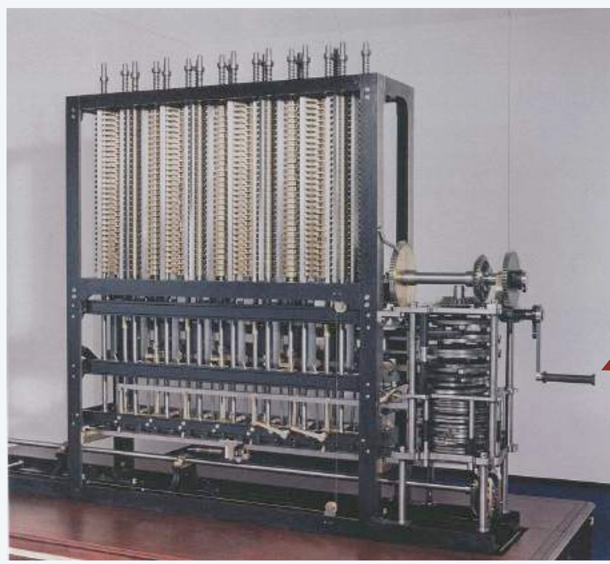
Analysis of Algorithms (Babbage, 1860s)



"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* (*1864*)

Analytic Engine



how many times do you have to turn the crank?

Analysis of Algorithms (Turing (!), 1940s)

"It is convenient to have a measure of the amount of work involved in a computing process, even though it be a very crude one. We may count up the number of times that various elementary operations are applied in the whole process . . ."

— *Alan Turing (1947)*

ROUNDING-OFF ERRORS IN MATRIX PROCESSES

By A. M. TURING

(*National Physical Laboratory, Teddington, Middlesex*)

[Received 4 November 1947]

SUMMARY

A number of methods of solving sets of linear equations and inverting matrices are discussed. The theory of the rounding-off errors involved is investigated for some of the methods. In all cases examined, including the well-known 'Gauss elimination process', it is found that the errors are normally quite moderate: no exponential build-up need occur.

Analysis of Algorithms (Knuth, 1960s)

Classical mathematics provides the necessary tools for understanding the performance of algorithms.

- Recurrence relations.
- Generating functions.
- Asymptotic analysis.

D. E. Knuth

BENEFITS:

Scientific foundation for AofA.

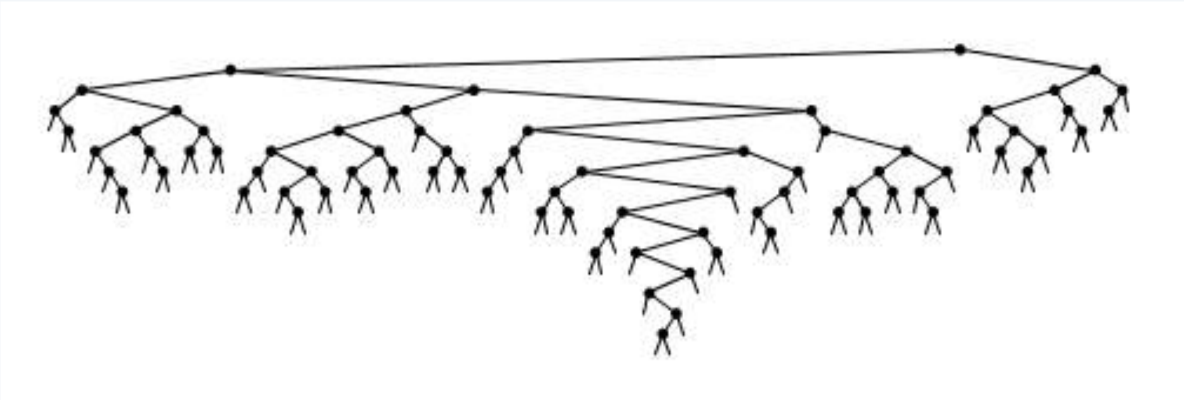
Can accurately predict performance and compare algorithms.

"If You Can Specify It, You Can Analyze It"

- Brief History
- **Analysis of Algorithms**
- Analytic Combinatorics
- Flajolet Collected Works
- New Directions

Analysis of algorithms: classic application

Q. How many bits needed to represent a binary tree with N internal nodes?



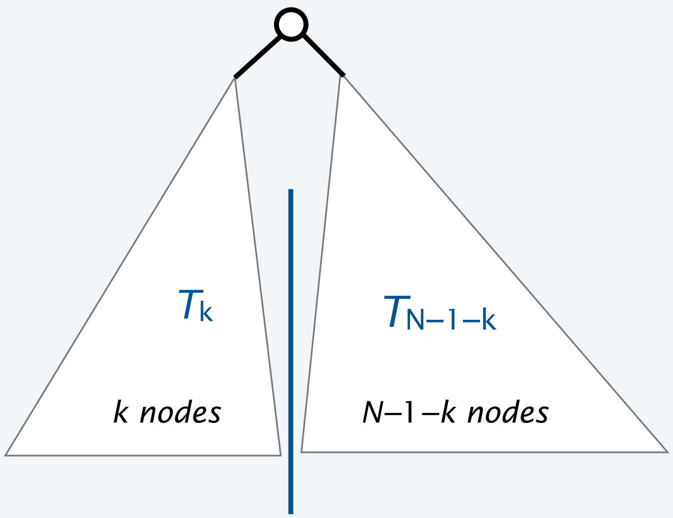
typical application: *data compression*

A. At least $\lg T_N$, where T_N is the number of binary trees with N internal nodes.

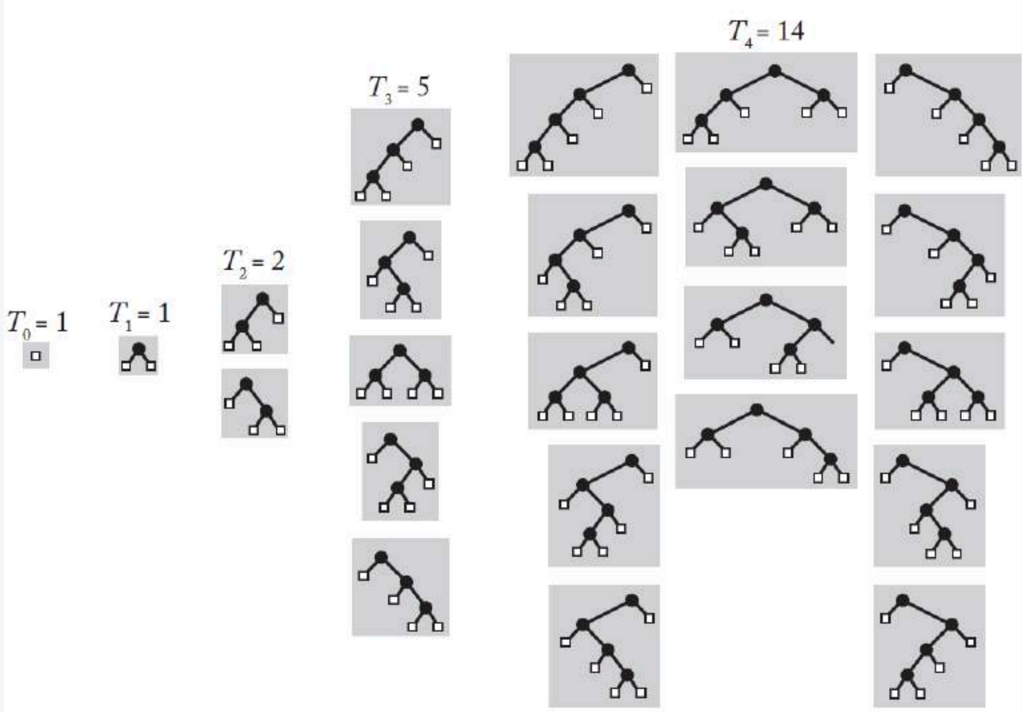
Q. How many binary trees with N internal nodes?

First step in classic AofA: Develop a recurrence relation

Q. How many binary trees with *N* internal nodes?



$$T_N = \sum_{0 \le k < N} T_k T_{N-1-k} + \delta_{N0}$$



Second step in classic AofA: Introduce a generating function

Generating functions have played a central role in scientific studies for centuries.

Rationale

- Provides concise representation of an infinite series with a single function.
- Studying the function provides information about the series.

Ordinary generating function (OGF) $$A(z) = \sum_{N \geq 0} A_N z^N$$

Exponential generating function (EGF) $$B(z) = \sum_{N \geq 0} \frac{B_N}{N!}$$

Abraham deMoivre
1667-1754

Second step in classic AofA: Introduce a generating function

Recurrence that holds for all N.

$$T_N = \sum_{0 \le k < N} T_k T_{N-1-k} + \delta_{N0}$$

GF

Multiply by z^N and sum.

$$T(z) \equiv \sum_{N \ge 0} T_N z^N = \sum_{N \ge 0} \sum_{0 \le k < N} T_k T_{N-1-k} z^N + 1$$

Switch order of summation

$$T(z) = 1 + \sum_{k \ge 0} \sum_{N > k} T_k T_{N-1-k} z^N$$

Change N to $N+k+1$

$$T(z) = 1 + \sum_{k \ge 0} \sum_{N \ge 0} T_k T_N z^{N+k+1}$$

Distribute.

$$T(z) = 1 + z\Big(\sum_{k \ge 0} T_k z^k\Big)\Big(\sum_{N \ge 0} T_N z^N\Big)$$

$$T(z) = 1 + zT(z)^2$$

recurrence yields *functional equation* satisified by GF

Third step in classic AofA: Extract coefficients

Functional GF equation. $$T(z) = 1 + zT(z)^2$$

Solve with quadratic formula. $$zT(z) = \frac{1}{2}(1 \pm \sqrt{1-4z})$$

Isaac Newton
1642-1726

Expand via binomial theorem. $$zT(z) = -\frac{1}{2}\sum_{N\ge 1}\binom{\frac{1}{2}}{N}(-4z)^N$$

Set coefficients equal $$T_N = -\frac{1}{2}\binom{\frac{1}{2}}{N+1}(-4)^{N+1}$$

Expand via definition. $$= -\frac{1}{2}\frac{\frac{1}{2}(\frac{1}{2}-1)(\frac{1}{2}-2)\ldots(\frac{1}{2}-N)(-4)^{N+1}}{(N+1)!}$$

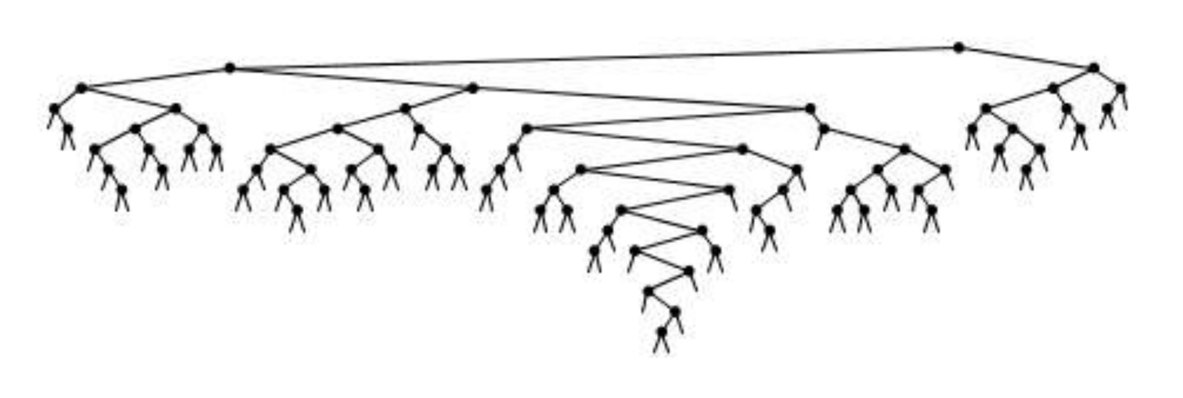
Distribute $(-2)^N$ among factors. $$= \frac{1\cdot 3\cdot 5\cdots(2N-1)\cdot 2^N}{(N+1)!}$$

Substitute (2/1)(4/2)(6/3)... for 2^N. $$= \frac{1}{N+1}\frac{1\cdot 3\cdot 5\cdots(2N-1)}{N!}\frac{2\cdot 4\cdot 6\cdots 2N}{1\cdot 2\cdot 3\cdots N}$$

Solution. $$T_N = \frac{1}{N+1}\binom{2N}{N}$$

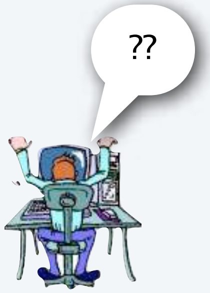
Analysis of algorithms: classic application

Q. How many bits needed to represent a binary tree with N internal nodes?



A. At least $\lg \frac{1}{N+1}\binom{2N}{N}$

Q. How many bits needed to represent a binary tree with 1000 internal nodes?



Fourth step in classic AofA: Asymptotics

Asymptotic approximations have played a central role in scientific studies for centuries.

Example: *Stirling's approximation* $\ln N! \sim N \ln N - N + \ln \sqrt{2\pi N}$

James Stirling
1692–1770

100	363.73	460.52	360.52	363.74
1000	5912.13	6907.76	5907.76	5912.13
10,000	82108.92	92103.40	82103.40	82108.93

Rationale

- Enables calculations of *precise* and *accurate* estimates for specific values.
- Provides *concise* representations using standard functions.
- *Asymptotic expansions* can increase accuracy with more terms.

Leonhard Euler
1707–1783

Pierre-Simon Laplace
1749–1827

Henri Poincaré
1854–1912

N. G. de Bruijn
1918–2012

Fourth step in classic AofA: Asymptotics

Solution. $$T_N = \frac{1}{N+1}\binom{2N}{N}$$

Apply exp-log. $$= \exp\bigl(\ln(2N!) - 2\ln N! - \ln(N+1)\bigr)$$

Stirling's approximation

$$\ln N! \sim N\ln N - N + \ln\sqrt{2\pi N}$$

Apply Stirling's approximation. $$\sim \exp\bigl(2N\ln(2N) - 2N + \ln\sqrt{4\pi N} - 2(N\ln(N) - N + \ln\sqrt{2\pi N}) - \ln N\bigr)$$

$$\ln\sqrt{4\pi N} - 2\ln\sqrt{2\pi N} = -\ln\sqrt{\pi N}$$

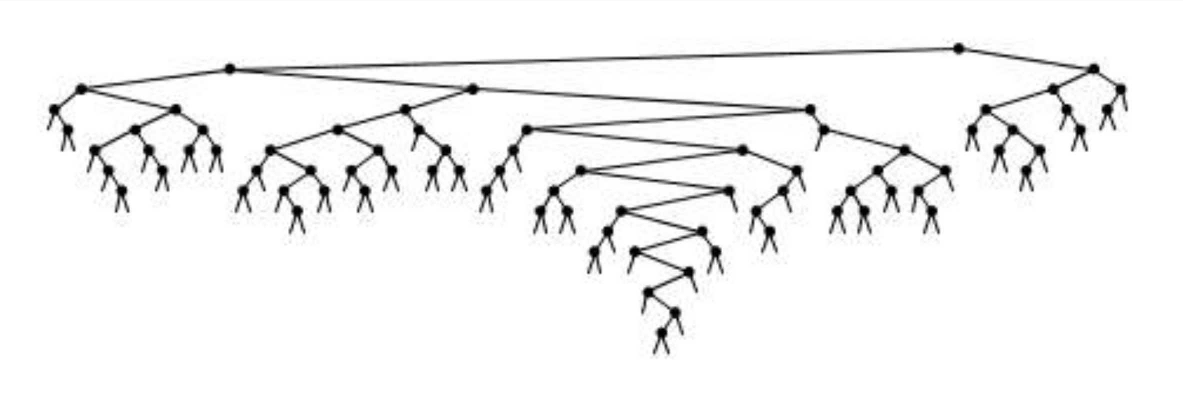
Simplify. $$= \exp\bigl(2N\ln 2 - \ln\sqrt{\pi N} - \ln N\bigr)$$

Undo exp-log. $$T_N \sim \frac{4^N}{\sqrt{\pi N^3}}$$

← easy to evaluate (in "standard scale")
can extend to any desired accuracy

Analysis of algorithms: classic application

Q. How many bits needed to represent a binary tree with N internal nodes?

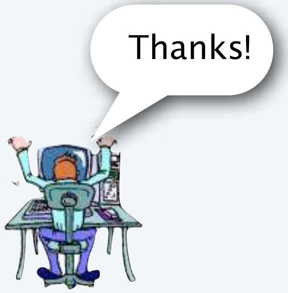


A. At least $\lg T_N \sim 2N - 1.5 \lg N$

Note 1: About 1985 for $N = 1000$

Note 2: Easy to do it with $2N$ bits

- Preorder traversal.
- Output 0 for internal nodes.
- Output 1 for external nodes.



Classic AofA: Summary

1. Develop recurrence relation.

$$T_N = \sum_{0 \le k < N} T_k T_{N-1-k} + \delta_{N0}$$

2. Derive GF equation.

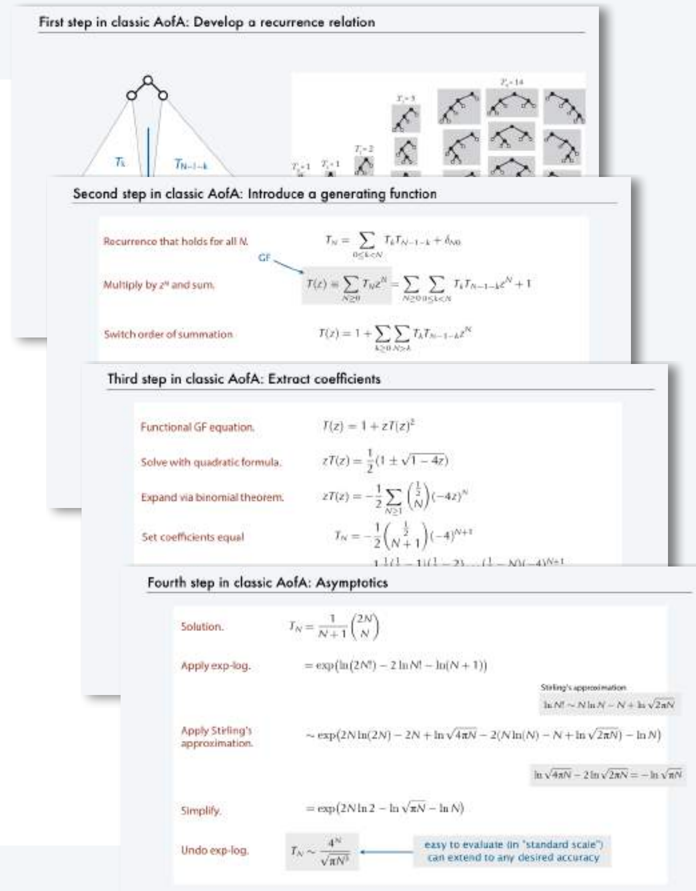
$$T(z) = 1 + zT(z)^2$$

3. Extract coefficients.

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

4. Develop approximation.

$$T_N \sim \frac{4^N}{\sqrt{\pi N^3}}$$



Challenge (1980): Efficiently teach math skills behind such derivations to CS students.

Analysis of Algorithms (Knuth, 1960s)

To analyze an algorithm:

- Develop a good implementation and a realistic input model.
- Determine the cost and execution frequency of each operation.
- Calculate the total running time: $\sum_q \text{frequency}(q) \times \text{cost}(q)$
- Run experiments to validate model and analysis.

the “scientific method”

D. E. Knuth

BENEFITS:

Scientific foundation for AofA.

Can predict performance and compare algorithms.

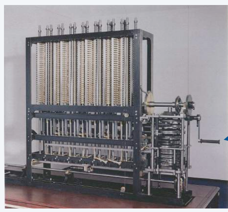
DRAWBACKS:

Model may be unrealistic.

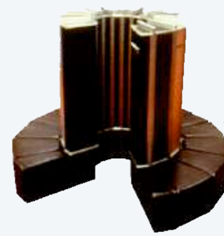
Significant classical math and excessive detail often needed for analysis.

AofA has played a critical role

in the development of our computational infrastructure *and the advance of scientific knowledge*



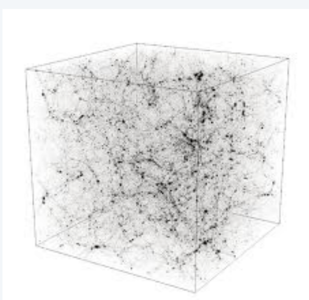
how many times to turn the crank?



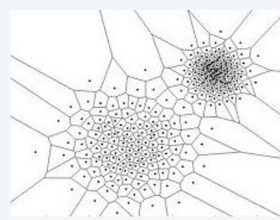
how long to sort random data for cryptanalysis preprocessing?

how long to compile my program?

how long to check that my VLSI circuit follows the rules?



how many bodies in motion can I simulate?



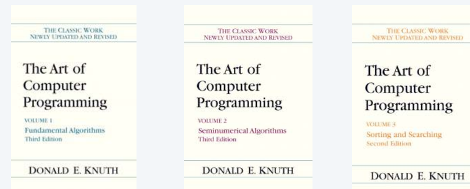
how quickly can I find clusters?

because the *scientific approach* enables performance predictions and algorithm comparisons

Genesis of “Analytic Combinatorics” (PF and RS, early 1980s)

Optimism and opportunity everywhere

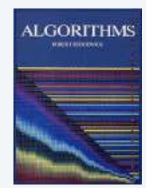
Knuth volumes 1-3



Search for generality

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

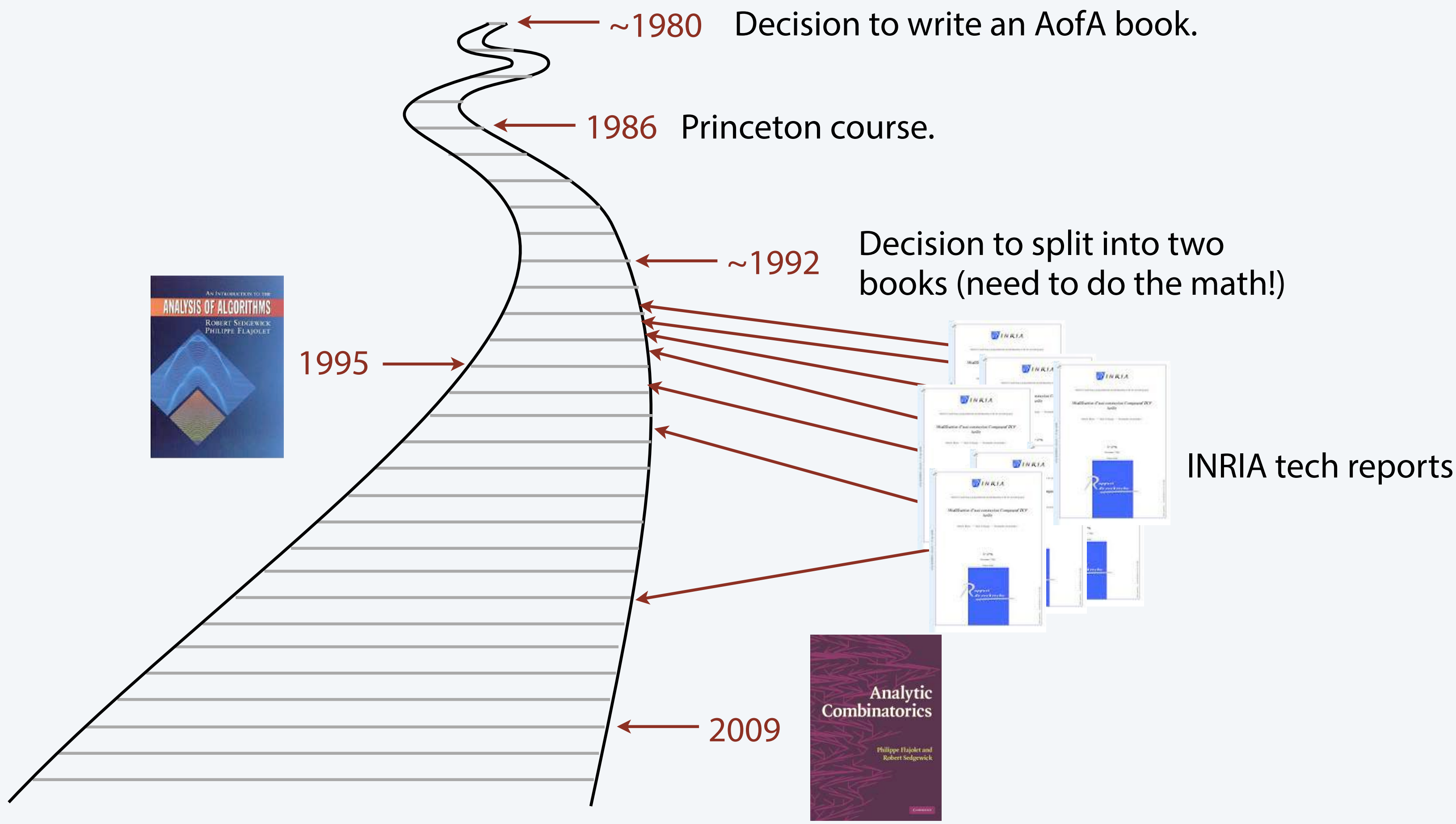
Algorithms for the masses



Teaching and research in AofA

Main motivation: Discover and teach basic methods and models to advance AofA.

Thirty years in the making



Analysis of Algorithms, 1995

Goal: Teach the mathematics needed for scientific study of the performance of computer programs.

Recurrences

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

Generating Functions

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

Asymptotics

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

Trees

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

Permutations

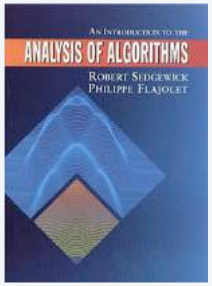
properties, representations, enumerations, inversions, cycles, extremal parameters

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

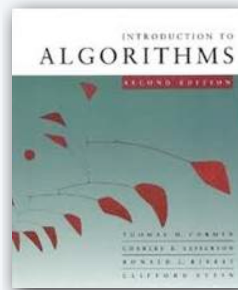
Theory of Algorithms (AHU, 1970s; CLRS, present day)

To address Knuth drawbacks:

- Analyze worst-case cost
  [takes model out of the picture].
- Use O-notation for upper bound
  [takes detail out of analysis].
- Classify algorithms by these costs.

Aho, Hopcroft and Ullman

Cormen, Leiserson, Rivest, and Stein



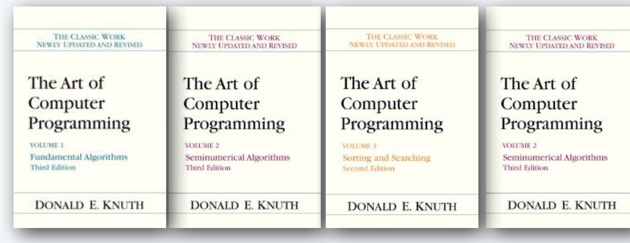
BENEFIT: Enabled a new Age of Algorithm Design.

DRAWBACK: Analysis is often *unsuitable* for scientific studies.
(An elementary fact that is often overlooked!)

Analytic combinatorics context

Drawbacks of Knuth approach:

- Model may be unrealistic.
- Analysis may involve excessive detail.

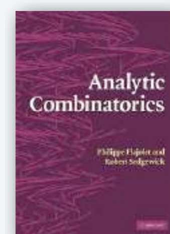


Drawbacks of AHU/CLRS approach:

- Worst-case performance may not be relevant.
- Cannot use O- upper bounds to predict or compare.

Analytic combinatorics can provide a basis for scientific studies.

- A calculus for developing models.
- Universal laws that encompass the detail in the analysis.



"If You Can Specify It, You Can Analyze It"

- Brief History
- Analysis of Algorithms
- **Analytic Combinatorics**
- Flajolet Collected Works
- New Directions

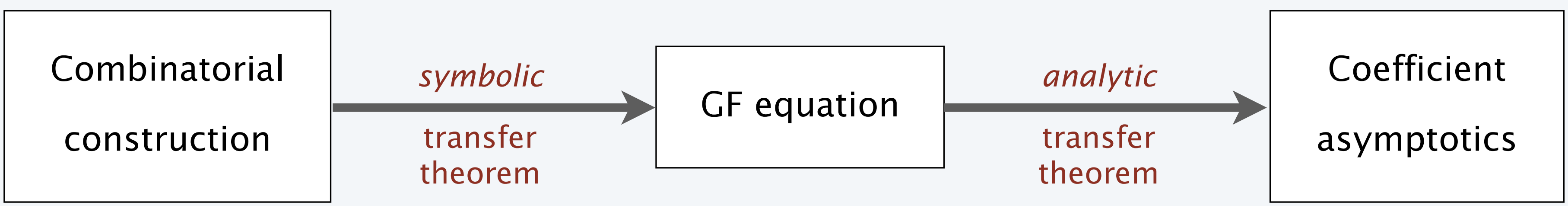
Analytic combinatorics

is a calculus for the quantitative study of large combinatorial structures.

Generating functions are the central object of study.

Basic process:

- Define a *combinatorial construction* that precisely specifies the structure
- Use a *symbolic transfer theorem* to derive a GF equation.
- Use an *analytic transfer theorem* to extract coefficient asymptotics.

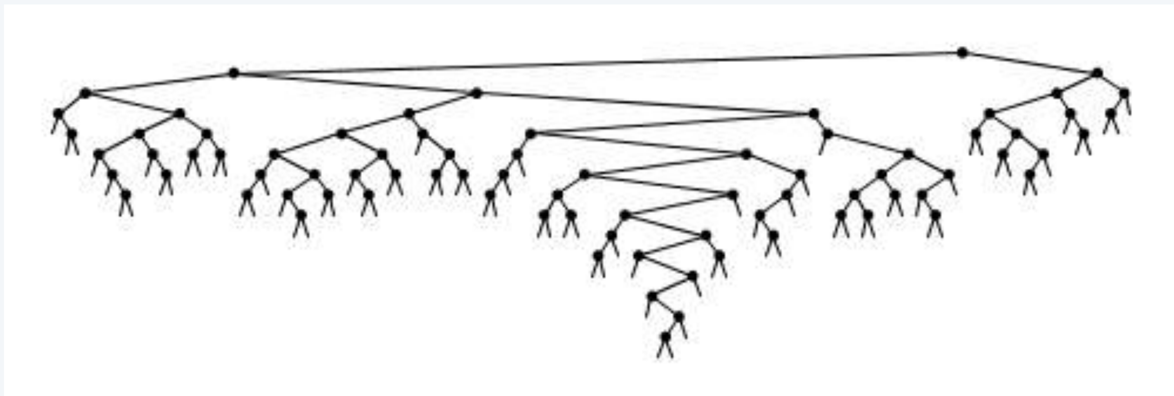


All three steps are often *immediate*.

Analytic combinatorics

is a calculus for the quantitative study of large combinatorial structures.

Ex: How many binary trees with N nodes?



$$T = E + Z \times T \times T$$

combinatorial construction

symbolic transfer theorem →

$$T(z) = 1 + zT(z)^2$$

GF equation

analytic transfer theorem →

$$T_N \sim \frac{4^N}{\sqrt{\pi N^3}}$$

coefficient asymptotics

First step in classic AC: Specify the class of objects being studied

using a *combinatorial construction* built from natural combinatorial operations.

Combinatorial constructions:

- Algebraic formulas built from natural *combinatorial operators.*
- Operands are *atoms* or other combinatorial constructions.
- Two cases: atoms are *unlabelled* (indistinguishable) or *labelled* (all different)

[Similar to formal languages, but with particular attention to ambiguity.]

Basic constructions (unlabelled classes)

Disjoint union	$A = B + C$
Cartesian product	$A = B \times C$
Sequence	$A = \mathrm{SEQ}(B)$

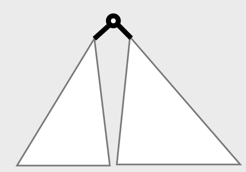
Example

$$T = E + Z \times T \times T$$

empty class

atom

"a binary tree is empty or a node and two binary trees"



Second step in classic AC: Introduce generating functions

and use *symbolic transfer theorems* to derive GF equation from construction.

Ordinary generating function

$$T(z) = \sum_{N\geq 0} T_N z^N = \sum_{t\in T} z^{|t|}$$

Basic transfer theorems (unlabelled classes)

Disjoint union	A = B + C	$A(z) = B(z) + C(z)$
Cartesian product	A = B × C	$A(z) = B(z)C(z)$
Sequence	A = SEQ(B)	$A(z) = \dfrac{1}{1 - B(z)}$

Example: Binary trees

Combinatorial class

T ≡ Set of all binary trees

Size function

$|t|$ ≡ Number of nodes in t

Counting sequence

T_N ≡ Number of trees with N nodes

Construction

T = E + Z × T × T

Transfer to GF equation

T = E + Z × T × T → $T(z) = 1 + zT(z)^2$

Generating functions

are the *key* to analytic combinatorics (but were controversial for some time)

"A property... is understood better, when one constructs a bijection... than when one calculates the coefficients of a polynomial whose variables have no particular meaning. The method of generating functions, which has had devastating effects for a century, has fallen into obsolescence, for this reason.

— Claude Berge, 1968

" Generating functions are the central objects of the theory, rather than a mere artifact to solve recurrences, as it is still often believed. "

— Philippe Flajolet, 2007

Third step in classic AC: Extract coefficients

using *analytic transfer theorems* based on viewing GF as complex function.

Fundamental transfer theorems *immediately provide* coefficient asymptotics.

Simple pole	$$[z^N]\frac{1}{(1-z/\rho)} = \rho^{-N}$$
Standard scale	$$[z^N]\frac{1}{(1-z/\rho)^\alpha} \sim \frac{N^{\alpha-1}}{\Gamma(\alpha)}\rho^{-N}$$
Standard scale (logarithmic)	$$[z^N]\frac{1}{(1-z)^\alpha}\ln\frac{1}{1-z} \sim \frac{N^{\alpha-1}}{\Gamma(\alpha)}\ln N$$

[long list, and growing]

$$[z^N]\sqrt{1-4z} \sim \frac{N^{-3/2}}{\Gamma(-1/2)}4^N = -2\frac{4^N}{\sqrt{\pi N^3}}$$

P. Flajolet and A. Odlyzko
Singularity analysis of generating functions.
SIAM J. Algebraic and Discrete Methods **3**, 1990.

and are effective even for approximations near singularities.

AofA vs. AC: Two ways to count binary trees

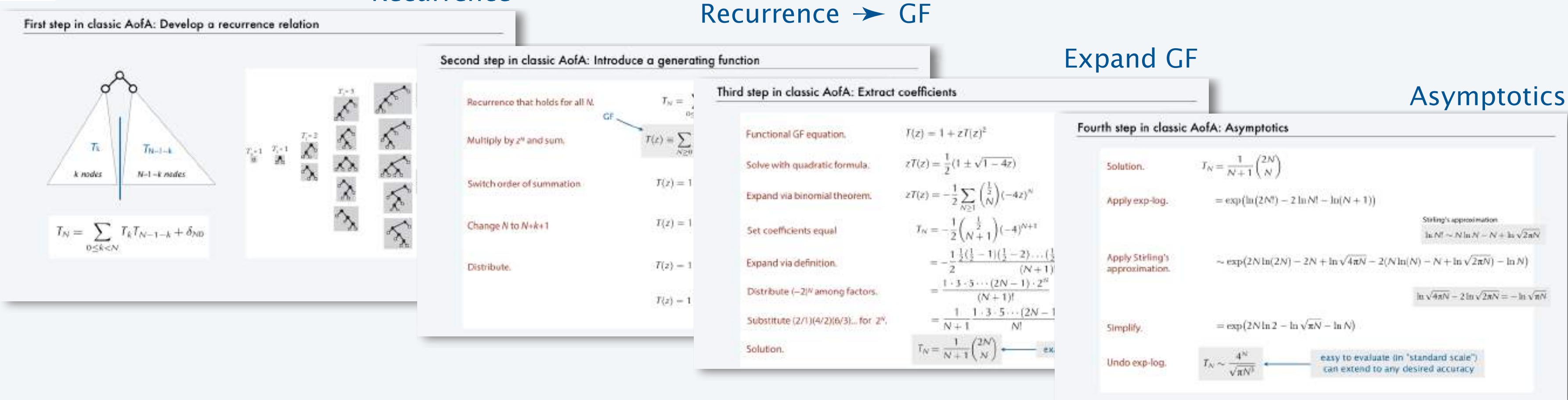
AofA

Recurrence

Recurrence → GF

Expand GF

Asymptotics



AC

T = E + Z × T × T

→

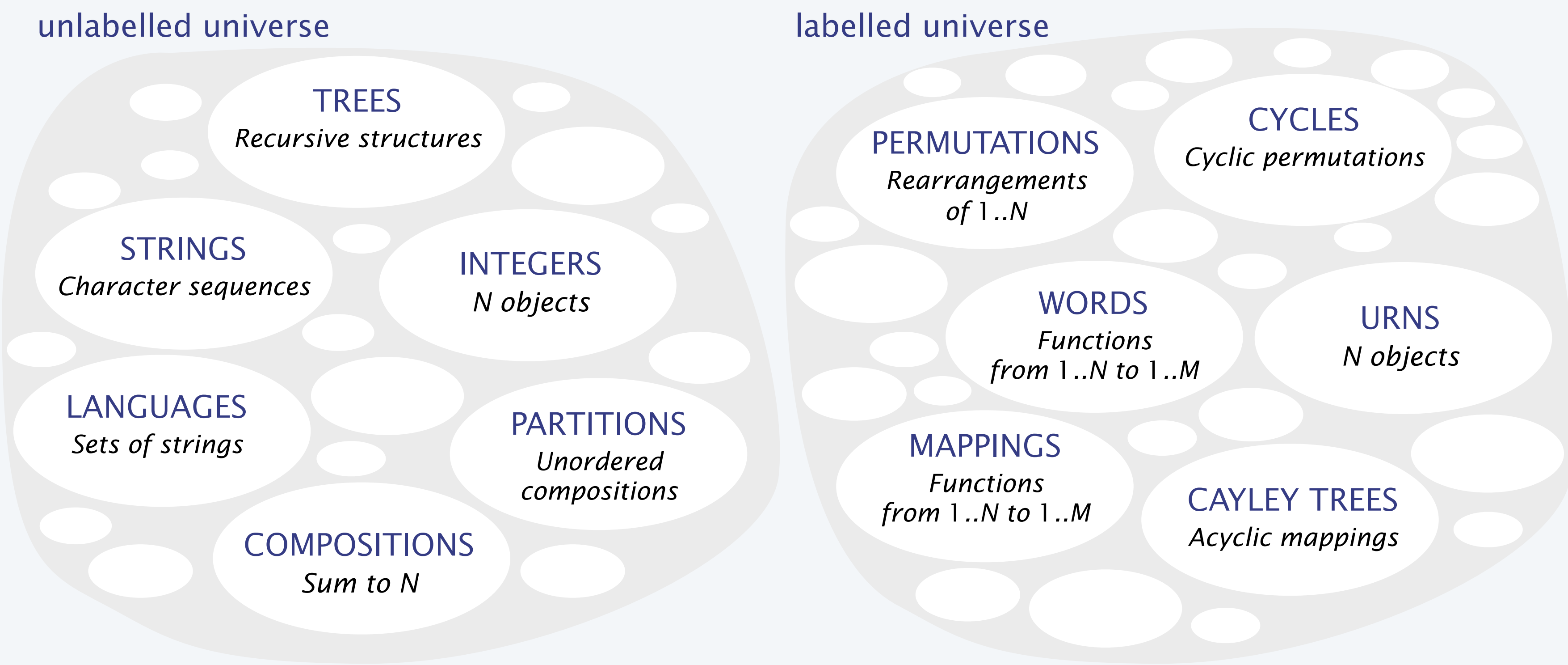
$$T(z) = 1 + zT(z)^2$$
$$= \frac{1}{2}(1 - \sqrt{1-4z})$$

→

$$T_N \sim \frac{4^N}{\sqrt{\pi N^3}}$$

“If you can specify it, you can analyze it”

AC is effective for a *broad variety* of combinatorial structures

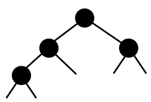
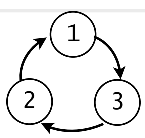


and is *fully extensible* (new constructions and transfers are being regularly discovered).

“If you can specify it, you can analyze it”

Combinatorial construction → GF equation → Coefficient asymptotics

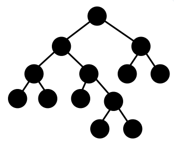
Elementary examples

Integers	●●●●●●●●	I = Z × SEQ(Z)	$I(z) = \frac{z}{1-z}$	$I_N = 1$ for $N > 0$
Strings	atttcgaa	W = SEQ(Z_0 + . . . + Z_{M-1})	$W_M(z) = \frac{1}{1-Mz}$	$W_{MN} = M^N$
Binary trees		T = E + ● × T × T	$T(z) = 1 + zT(z)^2$	$T_N \sim \frac{4^N}{\sqrt{\pi N^3}}$
Permutations	53724618	P = SEQ(Z)	$P(z) = \frac{1}{1-z}$	$P_N = N!$
Cycles		C = CYC(Z)	$C(z) = \ln \frac{1}{1-z}$	$C_N = (N-1)!$
Words	20010033	W_M = SEQ_M (SET(Z))	$W_M(z) = e^{Mz}$	$W_{MN} = M^N$

Sweet spot for AC: Variations on fundamental structures

Ex: Ordered (rooted plane) trees

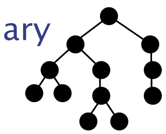
Binary



$$\mathsf{T} = \bullet + \bullet \times SEQ_{0,2}(\mathsf{T})$$

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

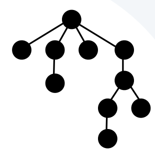
Unary-binary



$$\mathsf{M} = \bullet \times SEQ_{0,1,2}(\mathsf{M})$$

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

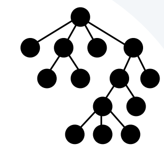
Ordered



$$\mathsf{G} = \bullet \times SEQ(\mathsf{G})$$

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

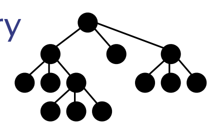
Bracketings



$$\mathsf{S} = \bullet \times SEQ_{>2}(\mathsf{S})$$

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

Ternary



$$\mathsf{T}^{[3]} = \bullet + \bullet \times SEQ_{0,3}(\mathsf{T}^{[3]})$$

$$T(z) = z(1 + T(z)^3)$$

Arbitrary restrictions

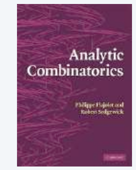
$$\mathsf{T}^{\Omega} = \bullet \times SEQ_{\Omega}(\mathsf{T}^{\Omega})$$

$$T^{\Omega}(z) = z\phi(T^{\Omega}(z))$$

$$\phi(u) \equiv \sum_{\omega \in \Omega} u^{\omega}$$

Universal laws

of sweeping generality are one hallmark of analytic combinatorics



Example: Context-free constructions

A *system* of combinatorial constructions

$$G_0 = OP_0(G_0, G_1, \ldots, G_t)$$
$$G_1 = OP_1(G_0, G_1, \ldots, G_t)$$
$$\ldots$$
$$G_t = OP_t(G_0, G_1, \ldots, G_t)$$

symbolic transfer

transfers to a *system* of GF equations

$$G_0(z) = F_0(G_0(z), G_1(z), \ldots, G_t(z))$$
$$G_1(z) = F_1(G_0(z), G_1(z), \ldots, G_t(z))$$
$$\ldots$$
$$G_t(z) = F_t(G_0(z), G_1(z), \ldots, G_t(z))$$

Grobner basis elimination

that reduces to a *single* GF equation

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

Drmota-Lalley-Woods theorem

that has an *explicit* solution

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

analytic transfer

that transfers to a *simple* asymptotic form

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

!!

One goal of modern research: Discover more universal laws.

Schemas

Combinatorial problems can be organized into broad schemas, covering infinitely many combinatorial types and governed by simple asymptotic laws.

Theorem. *Asymptotics of exp-log labelled sets.*
Suppose that a labelled set class **F** = SET_ϕ(**G**) is exp-log(α, β, ρ) with $G(z) \sim \alpha \log \frac{1}{1-z/\rho} + \beta$. Then $F(z) \sim e^\beta \left(\frac{1}{1-z/\rho}\right)^\alpha$ and $[z^N]F(z) \sim \frac{e^\beta}{\Gamma(\alpha)} \left(\frac{1}{\rho}\right)^N N^{1-\alpha}$

Theorem. *Asymptotics of supercritical sequences.* If **F** = SEQ(**G**) is a strongly aperiodic supercritical sequence class, then $[z^N]F(z) \sim \frac{1}{G'(\lambda)} \frac{1}{\lambda^{N+1}}$ where λ is the root of $G(\lambda) = 1$ in (0, ρ).

radius of convergence of $G(z)$

Theorem. If C is an irreducible context-free class, then its generating function $C(z)$ has a square-root singularity at its radius of convergence ρ. If $C(z)$ is aperiodic, then the dominant singularity is unique and $[z^N]F(z) \sim \frac{1}{\sqrt{\alpha\pi}} \left(\frac{1}{\rho}\right)^N N^{-3/2}$ where α is a computable real.

Theorem. If a simple variety of trees with GF $F(z) = z\phi(F(z))$ is λ-invertible (where λ is the positive real root of $\phi(u) = u\phi'(u)$) then $[z^N]F(z) \sim \frac{1}{\sqrt{2\pi\phi''(\lambda)/\phi(\lambda)}} \left(\phi'(\lambda)\right)^N N^{-3/2}$

Theorem. *Asymptotics of implicit tree-like classes.*
Suppose that **F** is an implicit tree-like class with associated GF $F(z) = \Phi(z, F(z))$ that is aperiodic and smooth-implicit(r, s), so that $G(r, s) = s$ and $G_w(r, s) = 1$. Then $F(z)$ converges at $z = r$ where it has a square-root singularity with $F(z) \sim s - \alpha\sqrt{1 - z/r}$ and $[z^N]F(z) \sim \frac{\alpha}{2\sqrt{\pi}} \left(\frac{1}{r}\right)^N N^{-3/2}$ where $\alpha = \sqrt{\frac{2r\Phi_z(r,s)}{\Phi_{ww}(r,s)}}$.

The discovery of such schemas and of the associated universality properties constitues the *very essence* of analytic combinatorics.

Analytic combinatorics at the next level

Combinatorial parameters are handled with MGFs, often leading to limit laws.

Complicated singularity structure leads to *oscillatory behavior* (like RS/PF formula in common).

GFs with no singularities require *saddle-point asymptotics*.

"If you can specify it, you can generate a *random structure*."

Analytic transfer theorems have *technical conditions* that need to be checked.

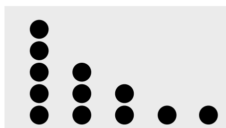
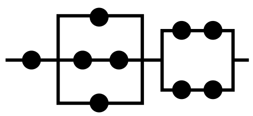
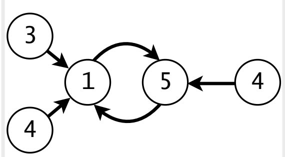
AofA involves understanding *transformations* from one combinatorial structure to another.

New types of *implicit GF functional equations* can arise.

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

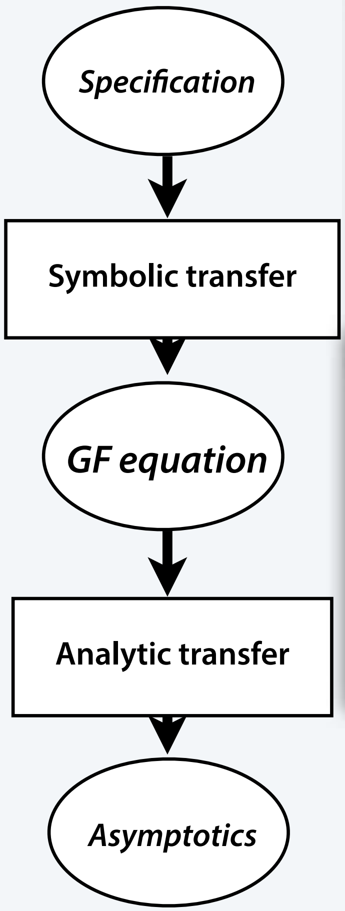
Combinatorial construction → GF equation → Coefficient asymptotics

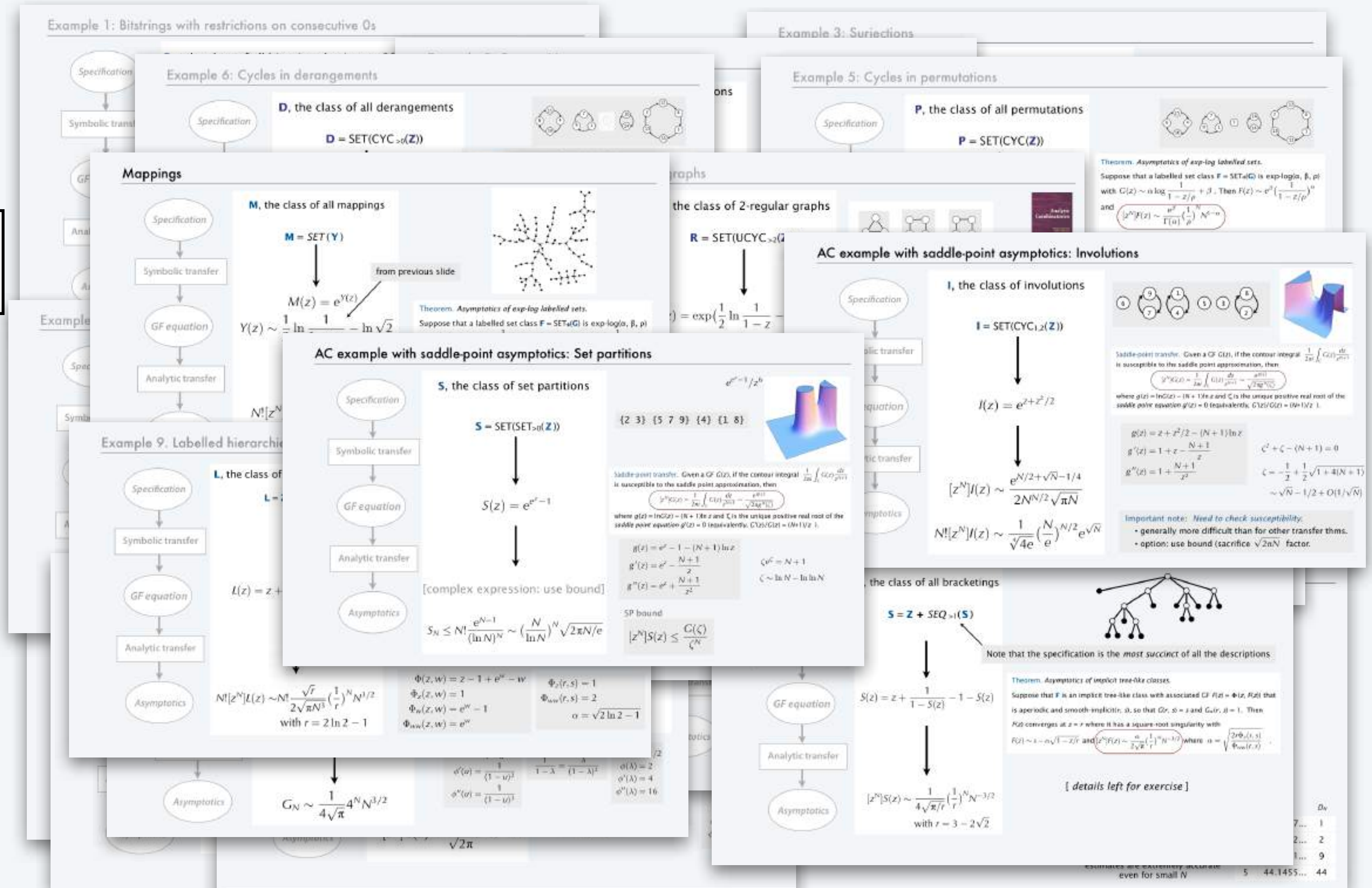
Representative examples

Partitions		P = MSET(I)	$P(z) = \frac{1}{(1-z)(1-z^2)(1-z^3)\ldots}$	$P_N \sim \frac{e^{\pi\sqrt{2N/3}}}{4N\sqrt{3}}$
Series-parallel networks		S = Z + ($SEQ_{>1}$ (S))	$S(z) = z + \frac{S(z)^2}{1-S(z)}$	$S_N \sim \frac{\rho^N}{4\sqrt{\rho\pi N^3}}$ $\rho = \frac{1}{3-\sqrt{8}}$
Surjections	20010021	R = SEQ ($SET_{>0}$ (Z))	$R(z) = \frac{1}{1-(e^z-1)} = \frac{1}{2-e^z}$	$R_N \sim \frac{N!}{2(\ln 2)^{N+1}}$
Components in mappings		C = Z × SET(C) Y = CYC(C)	$C(z) = ze^{C(z)}$ $Y(z) = \ln\frac{1}{1-C(z)}$	$Y_N \sim \frac{N^N\sqrt{\pi}}{\sqrt{2N}}$

[very long list, and growing]

“If you can specify it, you can analyze it”



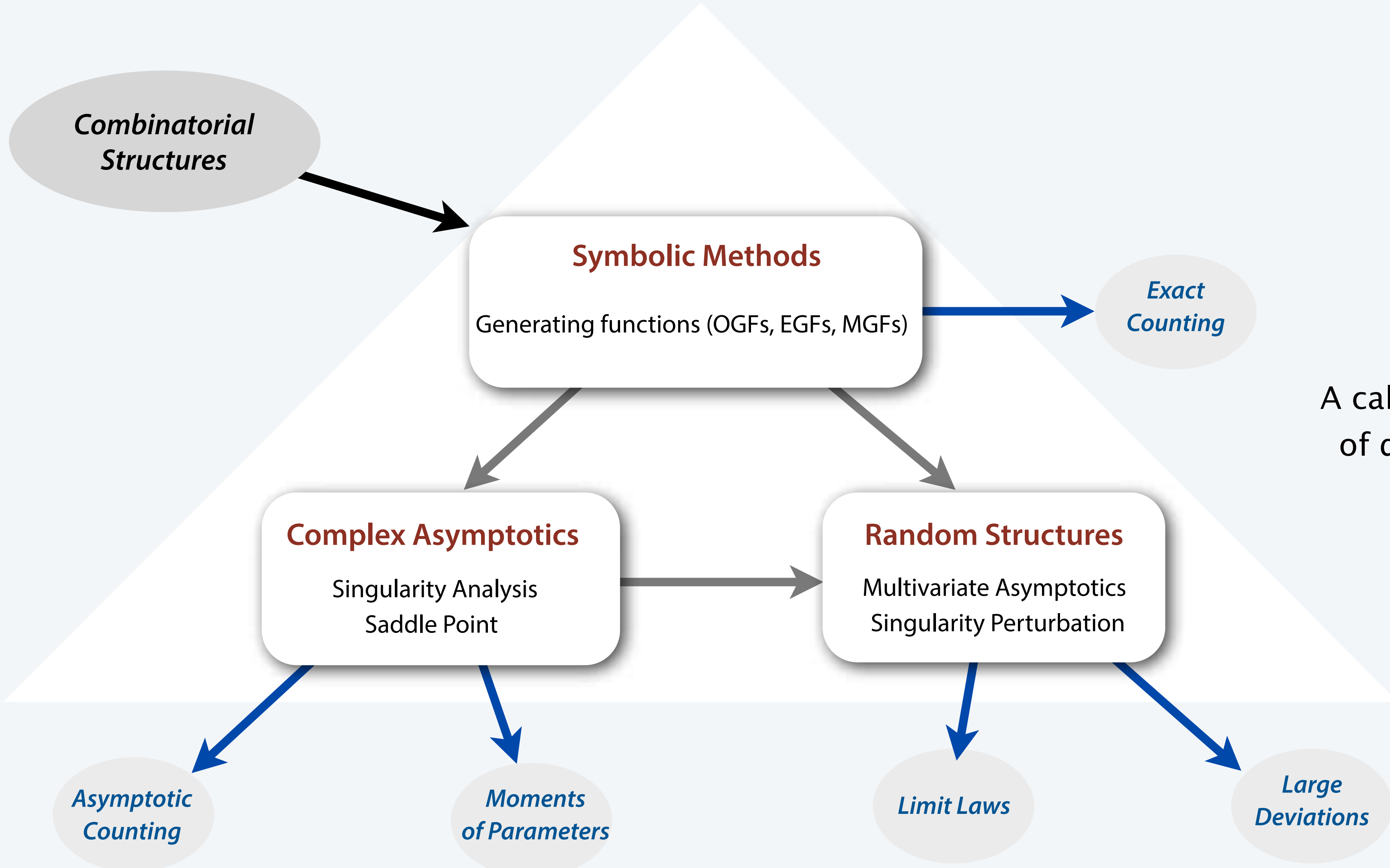


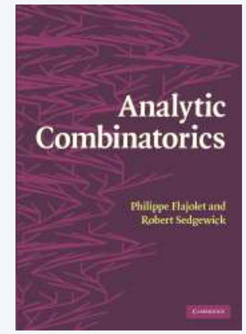
What is "Analytic combinatorics"?

[In case someone asks...]

Analytic combinatorics aims to enable precise quantitative predictions of the properties of large combinatorial structures. The theory has emerged over recent decades as essential both for the analysis of algorithms and for the study of scientific models in other disciplines, including statistical physics, computational biology, and information theory.

Analytic Combinatorics, 2009





✓

A calculus for the study of discrete structures

"If You Can Specify It, You Can Analyze It"

- Brief History
- Analysis of Algorithms
- Analytic Combinatorics
- **Flajolet Collected Works**
- New Directions

Collected Works of Philippe Flajolet

to be published by Cambridge University Press, 2014

Seven volumes

- Analytic Combinatorics
- Limit Laws and Dynamical Systems
- Text, Information Theory, and the Mellin Transform
- Trees and Graphs
- Combinatorial Structures
- Effective methods
- Theses and other writings

Strategy for this talk

- List of chapters in each volume.
- Discussion of a representative paper that is worth reading.
- Eye candy.

" If you read a paper of Philippe's, you will learn something. "

— H. K. Hwang, 2011

Volume One: Analytic Combinatorics

covers the basic research underlying the development of the field

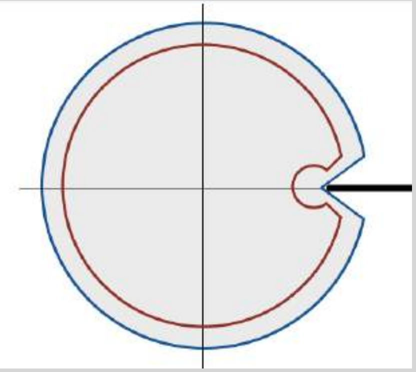
Chapter 1. *Analytic Combinatorics*

Chapter 2. *Singularity Analysis*

Chapter 3. *Thèse d'État (in English)*

Representative paper:

P. Flajolet and A. Odlyzko
Singularity analysis of generating functions.
SIAM J. Algebraic and Discrete Methods **3**, 1990.



Introduces fundamental complex-analytic transfer theorems

- Before this paper: "Folk theorems"
- After this paper: An effective calculus emerges.

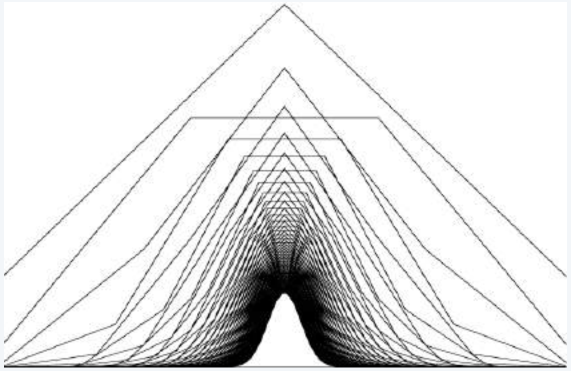
Volume Two: Limit Laws and Dynamical Systems

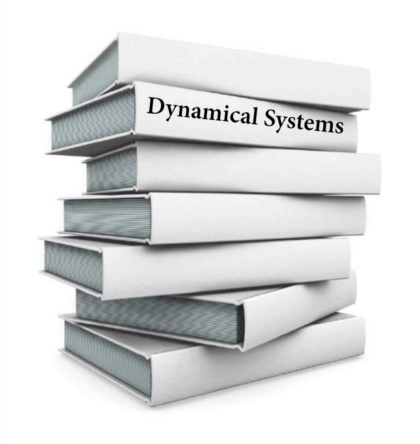
explores innovative approaches to the analysis of algorithms

Chapter 1. *Gaussian Limit Laws*

Chapter 2. *Airy Function*

Chapter 3. *Dynamical Systems*



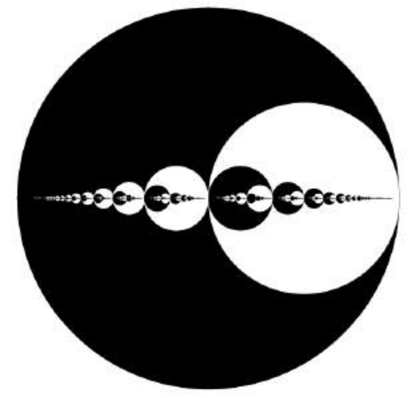


Representative paper:

J. Clément, P. Flajolet and B. Vallée

Dynamical sources in information theory: A general analysis of trie structures.

Algorithmica **29**, 2001.



Introduces models and analysis for string processing algorithms.

- Before this paper: Simplistic models.
- After this paper: Realistic models.

Volume Three: Text, Information Theory, and the Mellin Transform

addresses fundamental problems related to splitting processes.

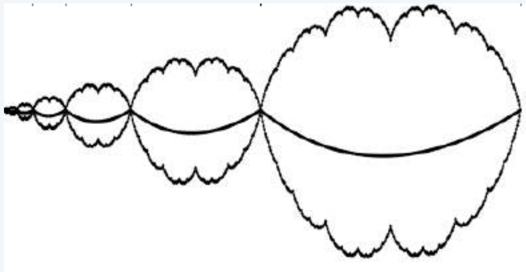
Chapters 1/2. *Text / Information Theory*

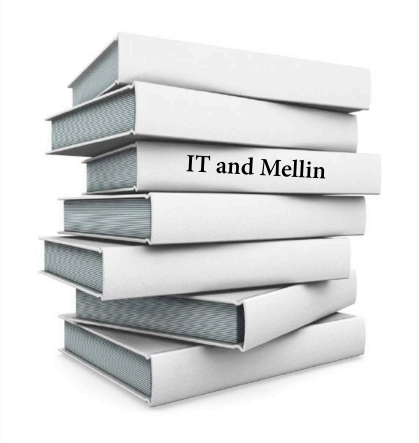
Chapter 3. *Tries & Digital Search Trees*

Chapter 4. *Mellin Transform*

Chapter 5. *Divide & Conquer*

Chapter 6. *Protocols*





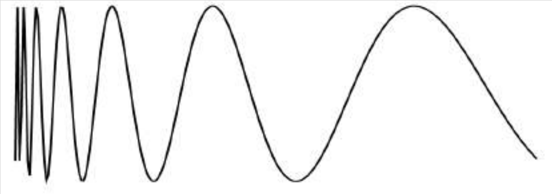
Representative paper:

P. Flajolet, X. Gourdon, and P. Dumas
Mellin transforms and Asymptotics: Harmonic Sums.
Theoretical Computer Science **144**, 1994.

Robert Mellin
1854–1933

Presents tools and techniques for analyzing recursive algorithms.

- Ties to classic analytic number theory.
- Volume 2 of *Analytic combinatorics*?



Volume Four: Trees and Graphs

illustrates the emergence of AC in the study of fundamental combinatorial structures.

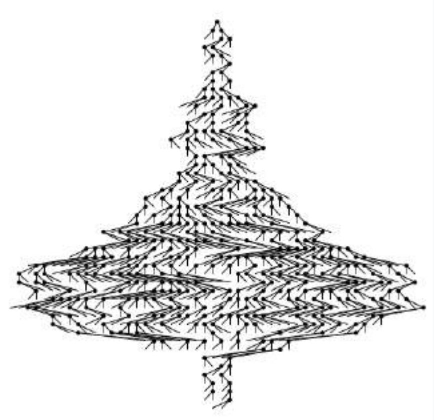
Chapter 1. *Term Trees*

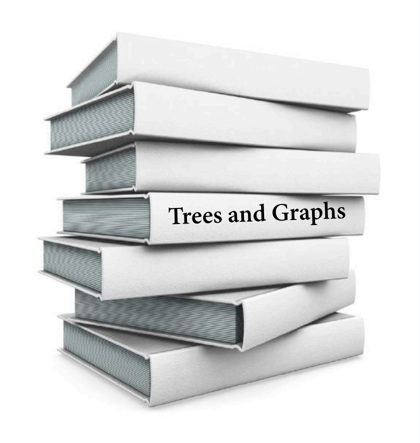
Chapter 2. *Height of Trees*

Chapter 3. *Search Trees*

Chapter 4. *Hashing*

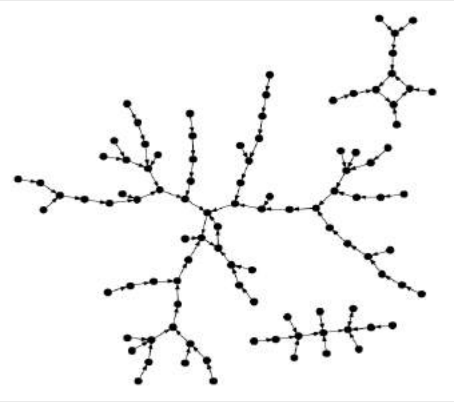
Chapter 5. *Random Graphs/Mappings*





Representative paper:

P. Flajolet and A. Odlyzko
Random mapping statistics.
in *Advances in Cryptology,* Springer-Verlag, 1990.



Gives full analysis of properties of random mappings.

- Poster child for utility of analytic combinatorics.
- Starting point for study of graph models *and* finite fields.

Volume Five: Combinatorial Structures

studies fundamental *and unusual* combinatorial structures of widespread applicability.

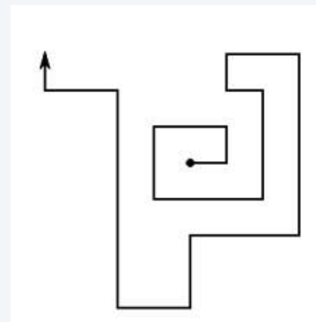
Chapter 1. *Languages*

Chapters 2/3. *Polynomials/Continued Fractions*

Chapter 4. *Random Walks and Lattice Paths*

Chapter 5. *Urns*

Chapters 6/7. *Number Theory/Register Function*

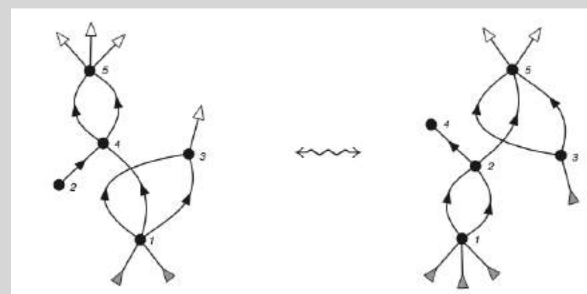


Representative paper:

P. Blasiak and P. Flajolet

Combinatorial Models of Creation-Annihilation.

Séminaire Lotharingien de Combinatoire **65**, 2011.



Surveys well-studied algebraic model from quantum physics.

- "Contains few new results."
- "Perhaps all known expansions in this orbit correspond to classic combinatorial models."

Volume Six: Effective Methods

covers practical and validated computational procedures.

Chapter 1. *Computer Algebra*

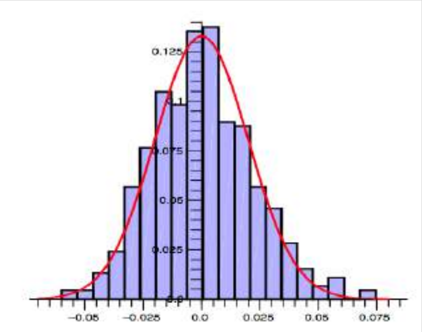
Chapter 2. *Automatic Analysis*

Chapter 3. *Random Generation and Simulation*

Chapter 4. *Approximate Counting*

Representative paper:

P. Flajolet, E. Fusy, O. Gandouet, and F. Meunier
Hyperloglog: analysis of a near-optimal cardinality estimation algorithm.
AofA 2007.



Culmination of field of research initiated by PF in 1985.

- Estimate cardinality in streams >> 10^9 to within 2% using ~1500 bytes.
- Method of choice in a broad variety of practical situations.

Volume Six: Effective Methods

covers practical and validated computational procedures.

Chapter 1. *Computer Algebra*

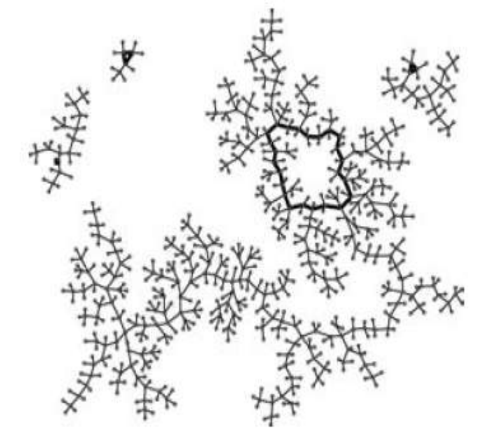
Chapter 2. *Automatic Analysis*

Chapter 3. *Random Generation and Simulation*

Chapter 4. *Approximate Counting*

Representative paper:

P. Duchon, P. Flajolet, G. Louchard, G. Schaeffer
Boltzmann Samplers for the Random Generation of Combinatorial Structures.
Combinatorics, Probability and Computing **13**, 2004.



Scalable algorithm for generating random structures.

- Immediate from combinatorial specification.
- *Linear* time.

Volume Seven: Theses and other writings

Chapter 1. *Ph.D. thesis*

Chapter 2. *Thèse d'État*

Chapter 3. *Short papers*

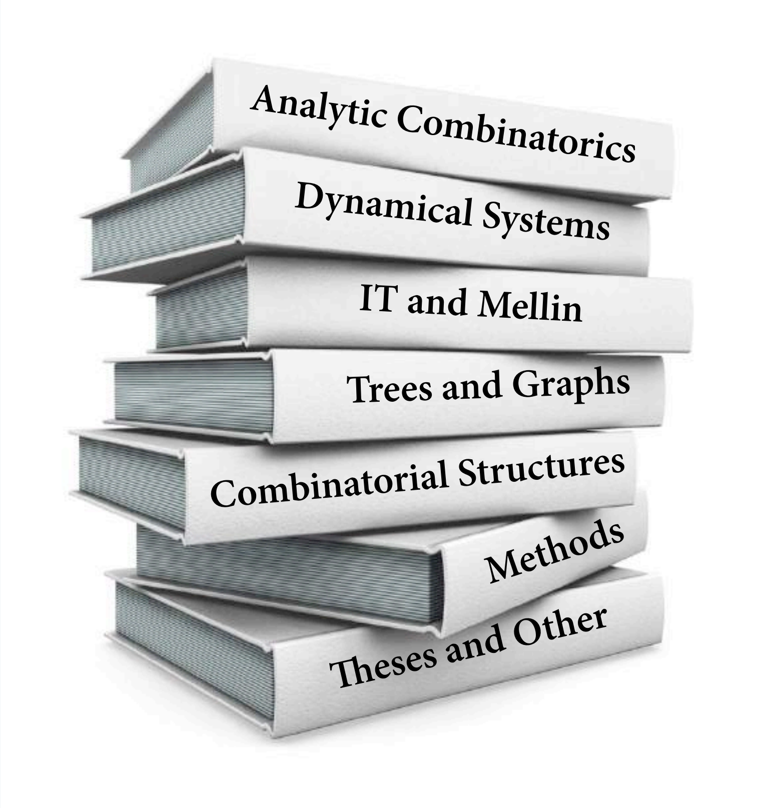
Chapter 4. *Notes for courses*

Chapter 5. *Reviews*

[mostly in French]

Standing on the shoulders of a giant

~185 papers
~5000 pages



Philippe Flajolet
1948–2011

Computer scientist

"Read Flajolet, *read Flajolet*, he is the master of us all."

[Adapted from Laplace's comment about Euler.]

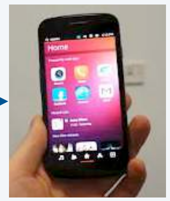
"If You Can Specify It, You Can Analyze It"

- Brief History
- Analysis of Algorithms
- Analytic Combinatorics
- Flajolet Collected Works
- **New Directions**

AofA/AC is more relevant than ever

because modern applications address huge and increasingly sophisticated problems

how much power left?



how long to download this TV series?

how long to search for patterns in this data?

how long to index the web?

how long to learn what ad to serve to this customer??

still demanding a *scientific approach* that enables performance predictions and algorithm comparisons

Example 1: Back to the Analysis of Algorithms

Q. Precise analysis of Divide-and-Conquer algorithms ?

A. Looks complicated. Use continous approximation and settle for order of growth.

Ex. Suppose that an algorithm attacks a problem of size n by dividing into α parts of size about n/β with extra cost $\Theta(n^\gamma(\log n)^\delta)$

Theorem. The solution to the recurrence

$$a_n = \underbrace{a_{n/\beta+O(1)} + a_{n/\beta+O(1)} + \ldots + a_{n/\beta+O(1)}}_{\alpha \text{ terms}} + \Theta(n^\gamma(\log n)^\delta)$$

is given by

$$a_n = \Theta(n^\gamma(\log n)^\delta) \qquad \text{when } \gamma < \log_\beta \alpha$$

$$a_n = \Theta(n^\gamma(\log n)^{\delta+1}) \qquad \text{when } \gamma = \log_\beta \alpha$$

$$a_n = \Theta(n^{\log_\beta \alpha}) \qquad \text{when } \gamma > \log_\beta \alpha$$

Example 1: Back to the Analysis of Algorithms

Q. Precise analysis of Divide-and-Conquer algorithms, *suitable for scientific studies* ?

A. YES! Classic AC.

$$T(n) = a_n + \sum_{1\le j\le m} b_j T(\lfloor h_j(x)\rfloor) + \sum_{1\le j\le m} \overline{b_j} T(\lceil \overline{h_j}(x)\rceil)$$

$T(n) = n + T(\lfloor n/2\rfloor) + T(\lceil n/2\rceil)$	$T(n) \sim n \lg n + \Psi(\lg n)$
$T(n) = n \log n + 2T(\lfloor n/2\rfloor) + 3T(\lceil n/6\rceil)$	$T(n) \sim cn^{1.402\ldots}$
$T(n) = \dfrac{n^2}{\log n} + 2T(\lfloor n/2\rfloor) + \dfrac{8}{9}T(\lfloor 3n/4\rfloor)$	$T(n) \sim cn^2 \ln\ln n$
$T(n) = 1 + pT(\lfloor pn+\delta\rfloor) + qT(\lceil qn-\delta\rceil)$	$T(n) \sim \dfrac{\log n - \alpha + \Psi(\log n)}{p\log(1/p) + q\log(1/q)}$

Ψ: 0 if $\dfrac{\log p}{\log q}$ is rational

M. Drmota and W. Szpankowski
A Master Theorem for Discrete Divide-and-Conquer Recurrences.
Journal of the ACM, to appear.

Example 2. Models for discrete structures in biochemistry

Q. Models for *RNA pseudoknot structures* ?

Critical for molecular function

Applications

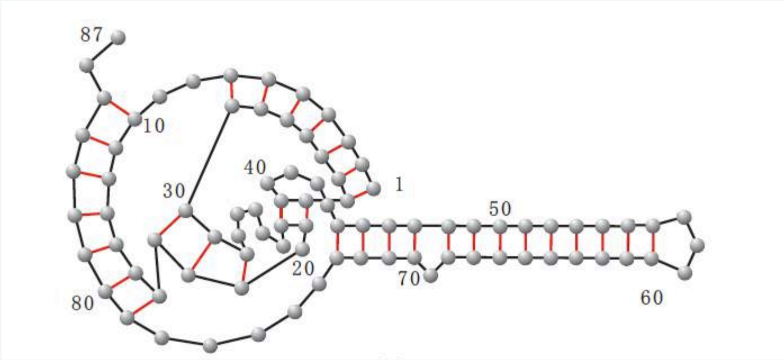
- catalytic cores of ribozymes
- telomerase activity
- programmed frameshifting

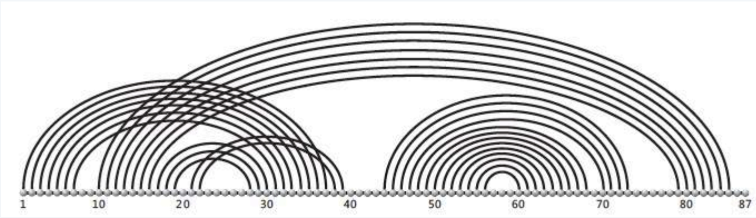
Issue. Problem is NP-complete. Need to consider restricted structures of various types.

Example 2. Models for discrete structures in biochemistry

Q. Model for *restricted RNA pseudoknot structures*?

A. YES! Need a new transfer theorem for MCGFs, but AC enables new research.





M. Nebel and F. Weinberg
Algebraic and Combinatorial Properties of Common RNA Pseudoknot Classes.
Journal of Computational Biology **10**, 2012.

Example 3: Random generation and modeling

Q. Models for *Software* ?

Applications

- model driven engineering
- ontology development
- abstract representations of knowledge

Example: QuickCheck

- combinator library
- written in Haskell
- generates test cases for test suites

Issue. Need better specifications of random structures

Example 3: Random generation and modeling

Q. *Meta*models for *Software* ?

A. YES! Use Boltzmann samplers.

$$
\begin{aligned}
Library &= Z \, * \, Book \, * \, Writer \\
Book &= void \\
Writer &= Z \\
Compilation &= Z \, * \, Book
\end{aligned}
$$

$$
\begin{aligned}
model &= package \\
package &= 0,01Z * Seq(packageableElement) \\
packageableElement &= package \mid class \mid association \\
class &= Z * Seq(property) * Seq(operation) * Seq(generalization) \\
generalization &= Z \\
property &= 3Z * (valueSpecification \mid \epsilon) \\
association &= Z \\
valueSpecification &= literalBoolean \mid literalNull \mid literalInteger \mid literalString \\
literalBoolean &= Z \\
literalNull &= Z \\
literalInteger &= 2Z \\
literalString &= Z \\
operation &= 2Z * Seq(parameter) \\
parameter &= 3Z * (valueSpecification \mid \epsilon)
\end{aligned}
$$

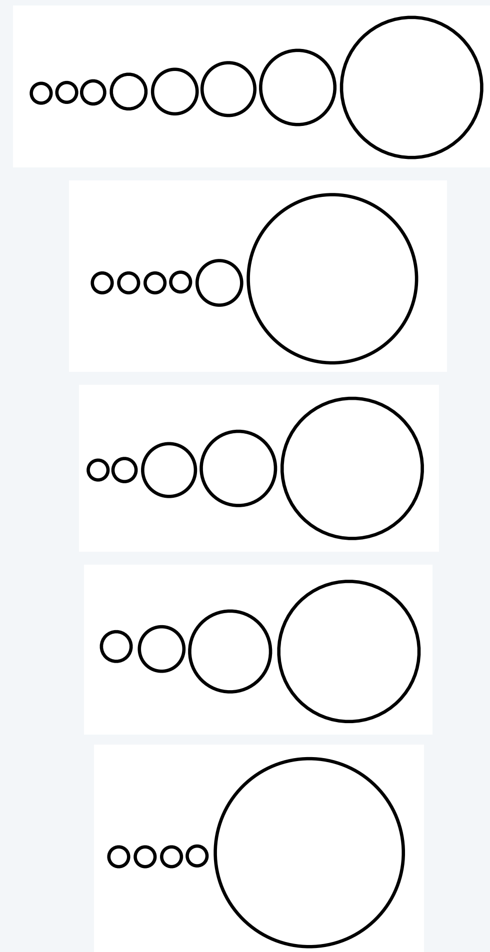
A. Mougenot, A. Darrasse, X. Blanc, M. Soria
Uniform Random Generation of Metamodel Instances.
Model Driven Architecture Foundations and Applications LNCS 5562, 2009.

Example 4: Finite fields

Q. Characterize polynomial factorizations over finite fields?

Applications

- design of cyclic redundancy codes
- partial fraction decompositions
- properties of elliptic curves
- building arithmetic public key cryptosystems
- computing discrete logarithms

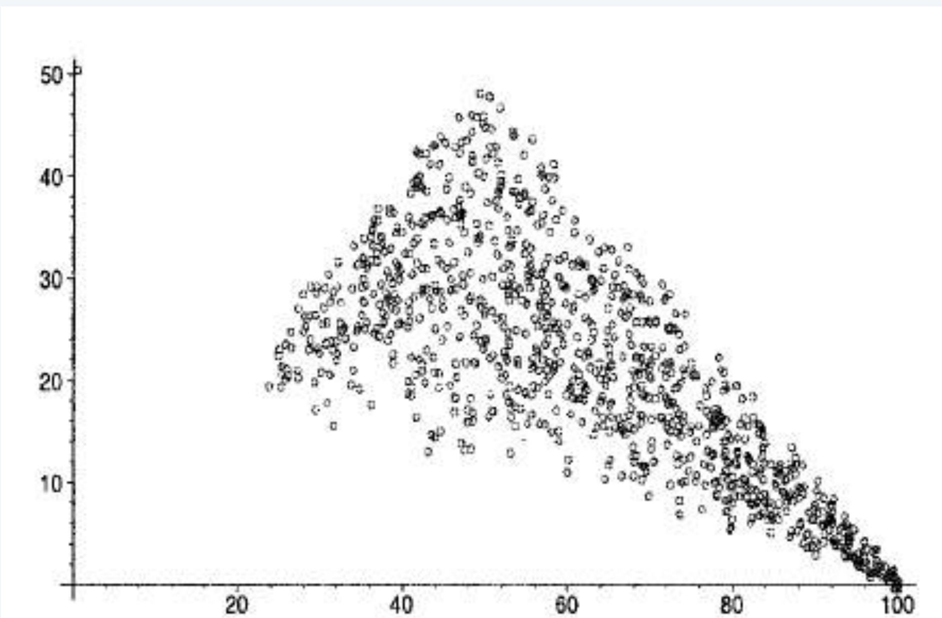


Issue. Need to understand sizes of factors to design efficient algs

Example 4: Finite fields

Q. Characterize polynomial factorizations over finite fields?

A. YES! Classic AC.

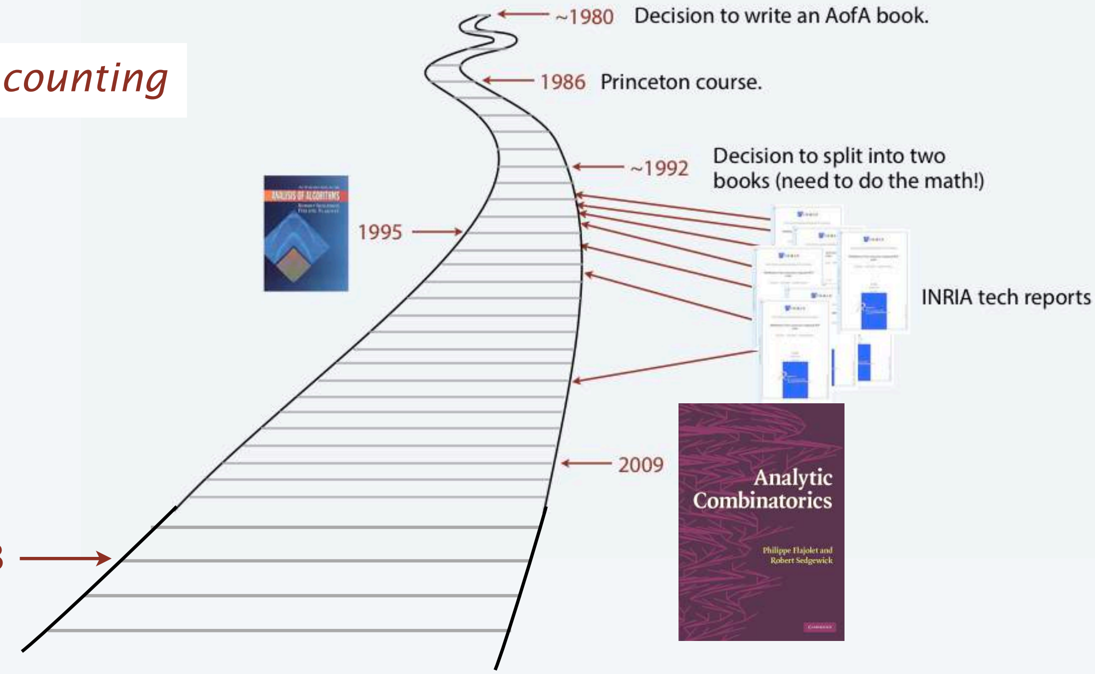


P. Flajolet, X. Gourdon and D. Panario
The complete analysis of a polynomial factorization algorithm over finite fields.
Journal of Algorithms **40**, 2001.

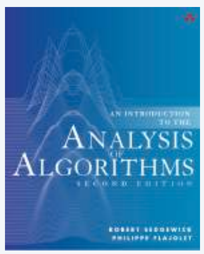
J. von zur Gathen, D. Panario and B. Richmond
Interval partitions and polynomial factorization.
Algorithmica **63**, 2012.

Dissemination

Thirty years in the making *and still counting*

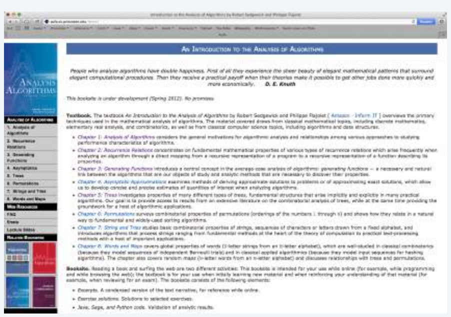


2nd edition
with intro to AC

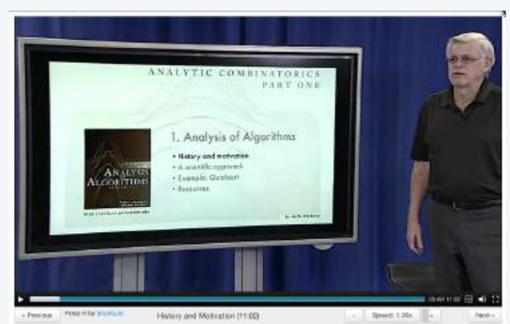


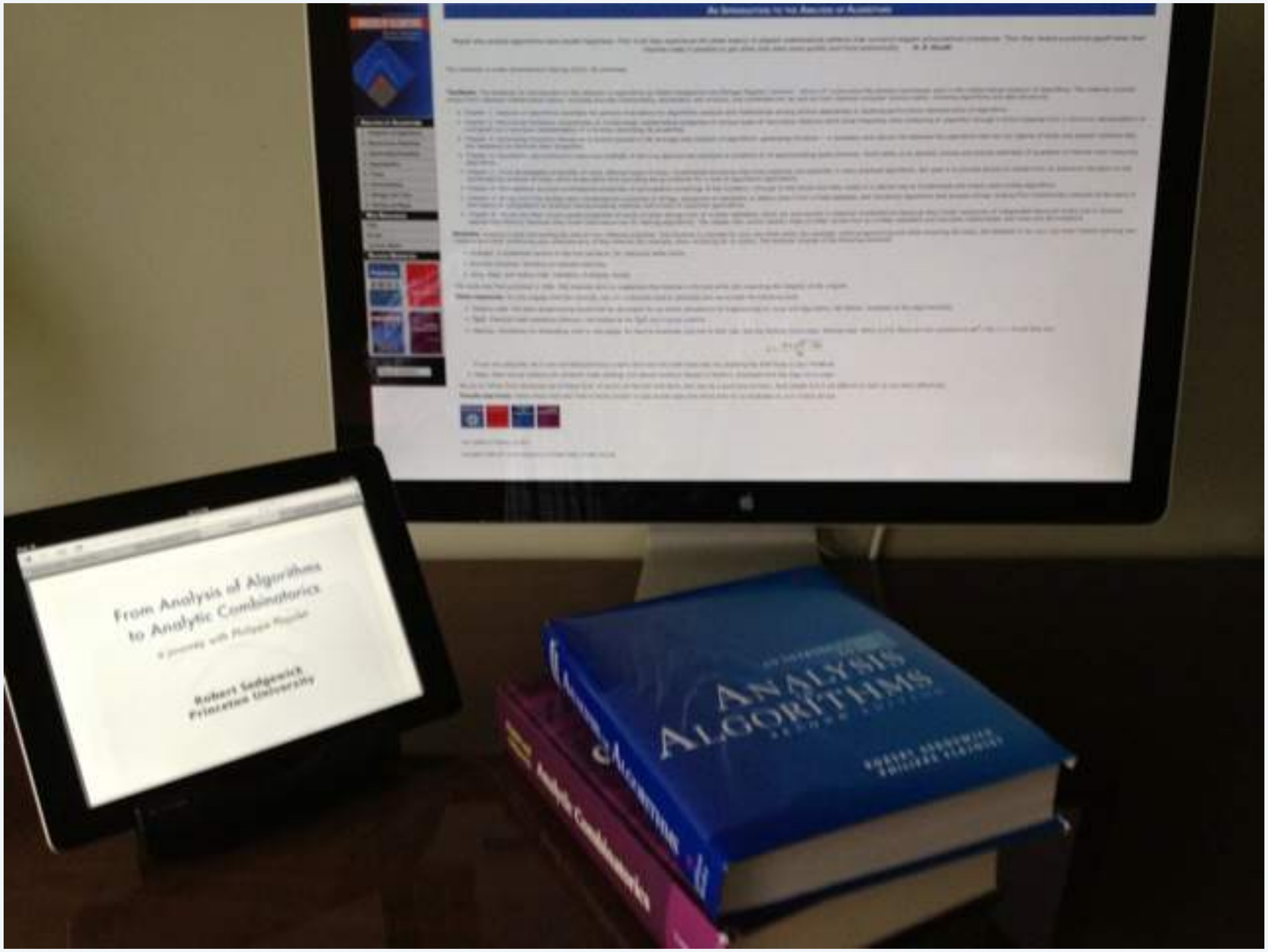
2013

web content



online courses



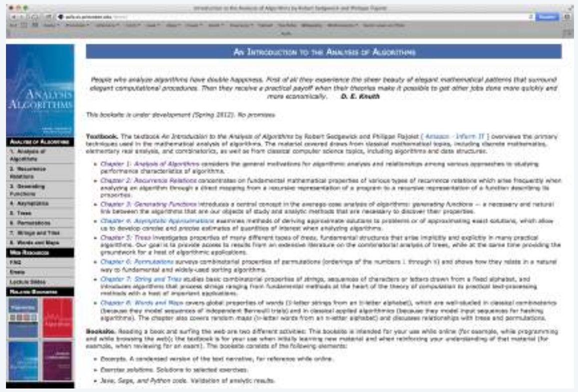

From Analysis of Algorithms
to Analytic Combinatorics
Robert Sedgewick
Princeton University
ANALYSIS
ALGORITHMS
Analytic Combinatorics

Dissemination

web content

`aofa.cs.princeton.edu`

- text digests
- code
- exercise solutions
- *lecture slides*



`ac.cs.princeton.edu`

(under construction)

online course

Analytic Combinatorics

10 lectures on AofA
10 lectures on AC

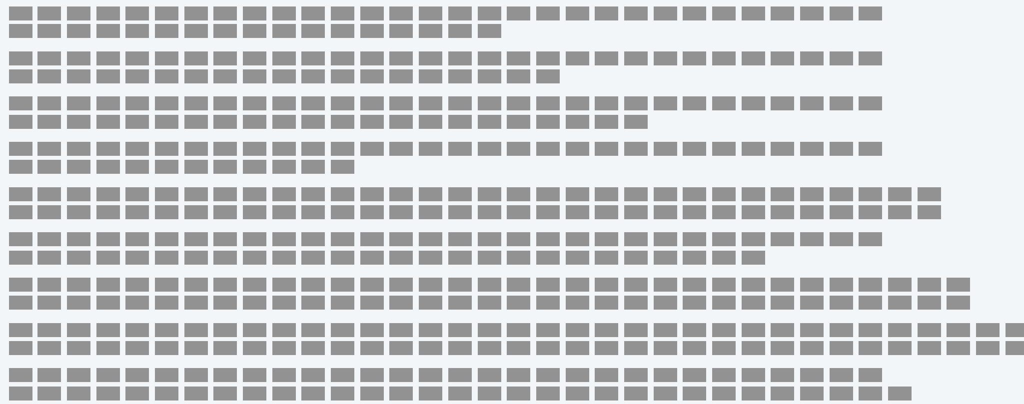
25,000+ registrants

Next offering:
September-December 2013

"Analytic Combinatorics" lectures

Part I: Analysis of Algorithms

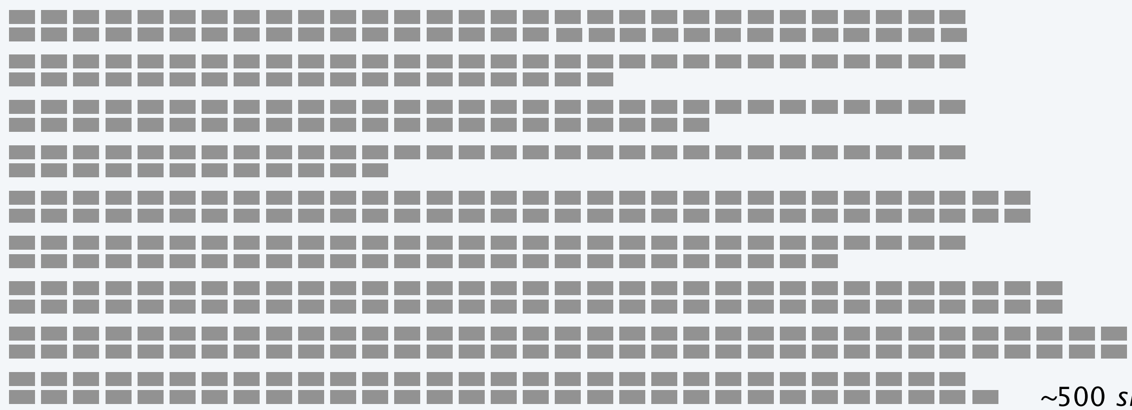
1. Introduction
2. Recurrences
3. Generating Functions
4. Asymptotic Analysis
5. Analytic combinatorics
6. Trees
7. Permutations
8. Strings and Tries
9. Words and Mappings



~500 *slides*
~50 *videos*

Part II: Analytic Combinatorics

1. Ordinary GFs
2. Exponential GFs
3. Bivariate GFs
4. Meromorphic Asymptotics
5. MA applications
6. Singularity Analysis
7. SA Applications
8. Saddle Point
9. Epilog



~500 *slides*
~50 *videos*

Just the beginning

" What is the most effective way to produce and disseminate knowledge with today's technology? How can we best structure what we know and learn so that students, researchers, and scholars of the future can best understand the work of today's researchers and scholars? "

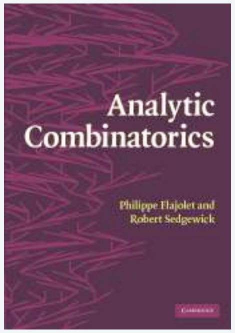
— Robert Sedgewick, 2007

If you can specify it, you can analyze it

Applications of analytic combinatorics

- patterns in random strings
- polynomials over finite fields
- quantum physics
- data compression
- geometric search
- combinatorial chemistry
- arithmetic algorithms
- planar maps and graphs
- probabilistic stream algorithms
- master theorem for divide-and-conquer
- bioinformatics
- automated testing

  . . .



A calculus for the study of discrete structures. ✓

"If You Can Specify It, You Can Analyze It"

the lasting legacy of Philippe Flajolet

Thanks, Philippe. It is a pleasure to be working with you!

Philippe Flajolet 1948–2011