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

the lasting legacy of Philippe Flajolet

Robert Sedgewick
Princeton University
"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!”
## Coming of age in CS (RS and PF generation)

<table>
<thead>
<tr>
<th>when we entered school</th>
<th>when we started work</th>
</tr>
</thead>
<tbody>
<tr>
<td>transistors</td>
<td>integrated circuits</td>
</tr>
<tr>
<td>punched cards</td>
<td>terminals</td>
</tr>
<tr>
<td>typewriter</td>
<td>word processing</td>
</tr>
<tr>
<td>Math</td>
<td>CS</td>
</tr>
</tbody>
</table>

A more profound change than PCs or the internet.
“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)
“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.
Classical mathematics provides the necessary tools for understanding the performance of algorithms.

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

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

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?

Typical application: data compression.
First step in classic AofA: Develop a recurrence relation

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

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

\[ B(z) = \sum_{N \geq 0} \frac{B_N}{N!} \]
Second step in classic AofA: Introduce a generating function

Recurrence that holds for all \( N \).

\[
T_N = \sum_{0 \leq k < N} T_k T_{N-1-k} + \delta_{N0}
\]

Multiply by \( z^N \) and sum.

\[
T(z) \equiv \sum_{N \geq 0} T_N z^N = \sum_{N \geq 0} \sum_{0 \leq k < N} T_k T_{N-1-k} z^N + 1
\]

Switch order of summation

\[
T(z) = 1 + \sum_{k \geq 0} \sum_{N > k} T_k T_{N-1-k} z^N
\]

Change \( N \) to \( N+k+1 \)

\[
T(z) = 1 + \sum_{k \geq 0} \sum_{N \geq 0} T_k T_{N} z^{N+k+1}
\]

Distribute.

\[
T(z) = 1 + z \left( \sum_{k \geq 0} T_k z^k \right) \left( \sum_{N \geq 0} T_N z^N \right)
\]

\[
T(z) = 1 + z T(z)^2
\]
Third step in classic AofA: Extract coefficients

Functional GF equation. 
\[ T(z) = 1 + zT(z)^2 \]

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

Expand via binomial theorem. 
\[ zT(z) = -\frac{1}{2} \sum_{N \geq 1} \left( \frac{1}{N} \right) (-4z)^N \]

Set coefficients equal 
\[ T_N = -\frac{1}{2} \left( \left( \frac{1}{N + 1} \right) (-4)^{N+1} \right) \]

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

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

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

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

Isaac Newton 
1642-1726
Analysis of algorithms: classic application

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

A. At least
\[
\log \left( \frac{1}{N + 1} \binom{2N}{N} \right)
\]

Q. How many bits needed to represent a binary tree with 1000 internal nodes?
Asymptotic approximations have played a central role in scientific studies for centuries.

Example: *Stirling’s approximation*  
\[ \ln N! \sim N \ln N - N + \ln \sqrt{2\pi N} \]

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.
Fourth step in classic AofA: Asymptotics

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

Apply exp-log. \[ = \exp \left( \ln(2N!) - 2 \ln N! - \ln(N + 1) \right) \]

Apply Stirling's approximation. \[ \sim \exp \left( 2N \ln(2N) - 2N + \ln(4\pi N) - 2(N \ln(N) - N + \ln(2\pi) - \ln N) \right) \]

Simplify. \[ = \exp \left( 2N \ln 2 - \ln \sqrt{\pi N} - \ln N \right) \]

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

Stirling's approximation:
\[ \ln N! \sim N \ln N - N + \ln(2\pi) \]

\[ \ln \sqrt{4\pi N} - 2 \ln \sqrt{2\pi} = -\ln \sqrt{\pi N} \]

easy to evaluate (in "standard scale") can extend to any desired accuracy
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. \[ T_N = \sum_{0 \leq k < N} T_k T_{N-1-k} + \delta_{N0} \]

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.
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.

**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

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

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

~1980  Decision to write an AofA book.

1986  Princeton course.

~1992  Decision to split into two books (need to do the math!)

1995

INRIA tech reports

2009
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
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.

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

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

**Benefit:** Enabled a new Age of Algorithm Design.
**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?

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

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

Combinatorial construction

Symbolic transfer theorem

GF equation

Analytic transfer theorem

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)

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Disjoint union</td>
<td>A = B + C</td>
</tr>
<tr>
<td>Cartesian product</td>
<td>A = B × C</td>
</tr>
<tr>
<td>Sequence</td>
<td>A = SEQ(B)</td>
</tr>
</tbody>
</table>

Example

\[
T = E + Z \times T \times T
\]

"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

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

Example: Binary trees

Combinatorial class

\[ T \equiv \text{Set of all binary trees} \]

Size function

\[ |t| \equiv \text{Number of nodes in } t \]

Counting sequence

\[ T_N \equiv \text{Number of trees with } N \text{ nodes} \]

Construction

\[ T = E + Z \times T \times T \]

Transfer to GF equation

\[
T(z) = 1 + zT(z)^2
\]
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.

<p>| |</p>
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
</table>
| **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) \[ [Z^N] \frac{1}{(1 - z)} \ln \frac{1}{1 - z} \sim \frac{N^{\alpha-1}}{\Gamma(\alpha)} \ln N \]  

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

P. Flajolet and A. Odlyzko
*Singularity analysis of generating functions.*

[long list, and growing ]

and are effective even for *approximations near singularities.*
AofA vs. AC: Two ways to count binary trees

### AofA

First step in classic AofA: Develop a recurrence relation

**Recurrence**

\[ T_0 = \sum_{n=1}^{N} T_n \]

Second step in classic AofA: Introduce a generating function

**Recurrence \(\Rightarrow\) GF**

**Expand GF**

Third step in classic AofA: Extract coefficients

Fourth step in classic AofA: Asymptotics

### AC

\[ T = E + Z \times T \times T \]

**Functional GF equation,**

\[ T(z) = T^2 + \frac{1}{z} \]

**Solve with quadratic formula,**

\[ T(z) = \frac{1}{2} (1 - \sqrt{1 - 4z}) \]

\[ T_N \sim \frac{4^N}{\sqrt{\pi N^3}} \]
"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”

<table>
<thead>
<tr>
<th>Elementary examples</th>
<th>Combinatorial construction</th>
<th>GF equation</th>
<th>Coefficient asymptotics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integers</td>
<td>$I = Z \times \text{SEQ}(Z)$</td>
<td>$I(z) = \frac{z}{1 - z}$</td>
<td>$I_N = 1$ for $N &gt; 0$</td>
</tr>
<tr>
<td>Strings</td>
<td>$W = \text{SEQ}(Z_0 + \ldots + Z_{M-1})$</td>
<td>$W_M(z) = \frac{1}{1 - Mz}$</td>
<td>$W_{MN} = M^N$</td>
</tr>
<tr>
<td>Binary trees</td>
<td>$T = E + \bullet \times T \times T$</td>
<td>$T(z) = 1 + zT(z)^2$</td>
<td>$T_N \sim \frac{4^N}{\sqrt{\pi N^3}}$</td>
</tr>
<tr>
<td>Permutations</td>
<td>53724618</td>
<td>$P(z) = \frac{1}{1 - z}$</td>
<td>$P_N = N!$</td>
</tr>
<tr>
<td>Cycles</td>
<td>$C = \text{CYC}(Z)$</td>
<td>$C(z) = \ln \frac{1}{1 - z}$</td>
<td>$C_N = (N - 1)!$</td>
</tr>
<tr>
<td>Words</td>
<td>20010033</td>
<td>$W_M(z) = e^{Mz}$</td>
<td>$W_{MN} = M^N$</td>
</tr>
</tbody>
</table>
Sweet spot for AC: Variations on fundamental structures

Ex: Ordered (rooted plane) trees

**Binary**

\[ T = \bullet + \bullet \times SEQ_{0,2}(T) \]

\[ T(z) = z(1 + T(z)^2) \]

**Ternary**

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

\[ T(z) = z(1 + T(z)^3) \]

**Ordered**

\[ G = \bullet \times SEQ(G) \]

\[ G(z) = \frac{z}{1 - G(z)} \]

**Arbitrary restrictions**

\[ T^{\Omega} = \bullet \times SEQ_{\Omega}(T^{\Omega}) \]

\[ T^{\Omega}(z) = z \phi(T^{\Omega}(z)) \]

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

**Unary-binary**

\[ M = \bullet \times SEQ_{0,1,2}(M) \]

\[ M(z) = z(1 + M(z) + M(z)^2) \]

**Bracketings**

\[ S = \bullet \times SEQ_{>2}(S) \]

\[ S(z) = z + \frac{S(z)^2}{1 - S(z)} \]
Universal laws

of sweeping generality are one hallmark of analytic combinatorics

Example: Context-free constructions

A system of combinatorial constructions

\[ G_0 = O P_0 (G_0, G_1, \ldots, G_t) \]
\[ G_1 = O P_1 (G_0, G_1, \ldots, G_t) \]
\[ \ldots \]
\[ G_t = O P_t (G_0, G_1, \ldots, G_t) \]

that reduces to a single GF equation

\[ G_0 (z) = F (G_0 (z), G_1 (z), \ldots, G_t (z)) \]

transfers to a system of GF equations

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

symbolic transfer

Groebner basis elimination

Drmota-Radley-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 $F = \text{SET}_\alpha(G)$ is $\exp-log(\alpha, \beta, \rho)$ with $G(z) \sim \alpha \log \frac{1}{1 - z^\rho / \rho}$ and $F(z) \sim \beta \left( \frac{1}{1 - z^\rho / \rho} \right)^\alpha$ and
\[
[z^n]F(z) \sim \frac{\beta}{\Gamma(\alpha)} \left( \frac{1}{\rho} \right)^n n^{-\alpha}.
\]

Theorem. Asymptotics of supercritical sequences. If $F = \text{SEQ}(G)$ is a strongly aperiodic supercritical sequence class, then
\[
[z^n]F(z) \sim \frac{1}{C'(\lambda)} \left( \frac{1}{\lambda^{1/2}} \right)^n n^{- \alpha}
\]
where $\lambda$ is the root of $G(\lambda) = 1$ in $(0, \rho)$.

Theorem. If $C$ is an irreducible context-free class, then its generating function $G(z)$ has a square-root singularity at its radius of convergence $\rho$. If $G(z)$ 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 $\alpha$ is a computable real.

Theorems. 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 $G_u(r, s) = 1$. Then $F(z)$ converges at $z = r$ where it has a square root singularity with
\[
F(z) \sim s - \alpha \sqrt{1 - z / r} \quad \text{and} \quad [z^n]F(z) \sim \frac{\alpha}{2 \sqrt{\pi}} \left( \frac{1}{r} \right)^{n-1/2} n^{-3/2}
\]
where $\alpha = \frac{2\sqrt{\pi}}{\Phi_{\text{tree}}(r,s)}$.

The discovery of such schemas and of the associated universality properties constitutes 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"

Representative examples

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

[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

- **Symbolic Methods**
  - Generating functions (OGFs, EGFs, MGFs)
  - **Exact Counting**

- **Complex Asymptotics**
  - Singularity Analysis
  - Saddle Point

- **Random Structures**
  - Multivariate Asymptotics
  - Singularity Perturbation

- **Asymptotic Counting**
- **Moments of Parameters**
- **Limit Laws**
- **Large Deviations**

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
Introduction to fundamental complex-analytic transfer theorems

Before this paper: "Folk theorems"

After this paper: An effective calculus emerges.

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.

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.

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.*

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.*


 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.*

Culmination of field of research initiated by PF in 1985.

- Estimate cardinality in streams $>> 10^9$ to within 2% using $\sim 1500$ bytes.
- Method of choice in a broad variety of practical situations.
Scalable algorithm for generating random structures.

- Immediate from combinatorial specification.
- Linear time.

Representative paper:

P. Duchon, P. Flajolet, G. Louchard, G. Schaeffer

*Boltzmann Samplers for the Random Generation of Combinatorial Structures.*

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]
"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

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 $\alpha$ parts of size about $n/\beta$ with extra cost $\Theta(n^\gamma(\log n)^\delta)$

**Theorem.** The solution to the recurrence

$$a_n = a_{n/\beta+O(1)} + a_{n/\beta+O(1)} + \cdots + a_{n/\beta+O(1)} + \Theta(n^\gamma(\log n)^\delta)$$

is given by

- $a_n = \Theta(n^\gamma(\log n)^\delta)$ when $\gamma < \log \beta \alpha$
- $a_n = \Theta(n^\gamma(\log n)^{\delta+1})$ when $\gamma = \log \beta \alpha$
- $a_n = \Theta(n^{\log \beta \alpha})$ 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.

\[
T(n) = a_n + \sum_{1 \leq j \leq m} b_j T(\lceil h_j(x) \rceil) + \sum_{1 \leq j \leq m} b_j T(\lceil h_j(x) \rceil)
\]

<table>
<thead>
<tr>
<th>( T(n) = n + T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) )</th>
<th>( T(n) \sim n \log n + \Psi(\log n) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T(n) = n \log n + 2T(\lfloor n/2 \rfloor) + 3T(\lceil n/6 \rceil) )</td>
<td>( T(n) \sim cn^{1.402...} )</td>
</tr>
<tr>
<td>( T(n) = \frac{n^2}{\log n} + 2T(\lfloor n/2 \rfloor) + \frac{8}{9} T(\lfloor 3n/4 \rfloor) )</td>
<td>( T(n) \sim cn^2 \ln \ln n )</td>
</tr>
<tr>
<td>( T(n) = 1 + pT(\lfloor pn + \delta \rfloor) + qT(\lceil qn - \delta \rceil) )</td>
<td>( T(n) \sim \frac{\log n - \alpha + \Psi(\log n)}{p \log(1/p) + q \log(1/q)} )</td>
</tr>
</tbody>
</table>

M. Drmota and W. Szpankowski

*A Master Theorem for Discrete Divide-and-Conquer Recurrences.*

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.*

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. Metamodels for Software?

A. YES! Use Boltzmann samplers.

A. Mougenot, A. Darrasse, X. Blanc, M. Soria

Uniform Random Generation of Metamodel Instances.

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.

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

web content

online courses

2013

~1980 Decision to write an AoA book.

1986 Princeton course.

~1992 Decision to split into two books (need to do the math!)

1995 INRIA tech reports

2009
Dissemination

web content

aofa.cs.princeton.edu

text digests
code
exercise solutions
lecture slides

analytic combinatorics

10 lectures on AofA
10 lectures on AC

25,000+ registrants

Next offering:
September-December 2013

ac.cs.princeton.edu
(under construction)
"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

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

~500 slides
~50 videos
“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

...
"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