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

Data Movement in Odd-Even Merging by Robert Sedgewick

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

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

\[
\frac{2k\pi - \log 2}{\log 2} \Gamma\left(\frac{k\pi}{\log 2}\right) \zeta\left(\frac{2k\pi}{\log 2}\right)
\]
Coming of Age in 1968
Coming of age in CS (RS and PF generation)

when we entered school

ad men

when we started work

hippies
Coming of age in CS (RS and PF generation)

**when we entered school**

transistors
(a physical device for every switch)

**when we started work**

integrated circuits
(Moore’s Law)
Coming of age in CS (RS and PF generation)

when we entered school

punched cards
(one run per day)

timesharing terminal
(always connected)

when we started work
Coming of age in CS (RS and PF generation)

when we entered school

when we started work

typewritten papers
(a week to a month turnaround)

word processing
(troff and TeX around the corner)
Coming of age in CS (RS and PF generation)

when we entered school

**Math**
- multivariate calculus
- topology
- ODEs, PDEs
- complex analysis
- abstract algebra
- probability
- statistics
- number theory

when we started work

**CS**
- compilers
- algorithms
- data structures
- computer graphics
- algorithms
- numerical analysis
- programming languages
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.
Analysis of Algorithms
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

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 (early 1980s)

Optimism and opportunity

Knuth volumes 1-3

Search for generality

Algorithms for the masses

Teaching and research in AofA

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

Main idea:
Teach the basics so CS students can get started on AofA.
Analysis of algorithms: classic example I

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.

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

Then introduce a generating function.

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

Multiply both sides by $z^N$ and sum to get an equation that we can solve and expand to get coefficients that we can approximate.

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

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

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

$$C_N \sim 2N \ln N$$  \textit{Euler-MacLaurin summation}

Note: Analyzing a property of permutations, not counting trees.
A binary tree is a node connected to two binary trees. How many binary trees with \( N \) nodes?

Develop a recurrence relation.

\[
B_N = \sum_{0 \leq k < N} B_k B_{N-1-k} \quad B_0 = 0
\]

Then introduce a generating function.

\[
B(z) = \sum_{k \geq 0} z^k
\]

Multiply both sides by \( z^N \) and sum to get an equation that we can solve algebraically.

\[
B(z) = 1 + zB(z)^2
\]

Quadratic equation

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

Binomial theorem

and expand to get coefficients that we can approximate.

\[
B_N = \frac{1}{N+1} \binom{2N}{N}
\]

\[
B_N \sim \frac{4^N}{N\sqrt{\pi N}}
\]

Stirling’s approximation

Challenge: Efficiently teach basic math skills behind such derivations.
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

1995

2009
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.

input model

Algorithm

Asymptotic estimate of running time

N floats

BST construction

~c N \ln N seconds
Analytic Combinatorics
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,

which directly maps to a GF equation

that we can manipulate algebraically

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

Challenge: Develop an effective calculus for such derivations.

Note: Construction for BSTs is not so simple.
Analytic combinatorics overview

To analyze properties of a large combinatorial structure:

1. Use the **symbolic method**
   - Define a *class* of combinatorial objects.
   - Define a notion of *size* (and associated generating function)
   - Use standard operations to develop a *specification* of the structure.

Result: A direct derivation of a *GF equation* (implicit or explicit).

2. Use **complex asymptotics** to estimate growth of coefficients.
   - No need for explicit solution.
   - General “transfer theorems” provide immediate results.

Result: *Asymptotic estimates* that quantify the desired properties.
Analytic combinatorics overview

A. SYMBOLIC METHOD

1. OGFs
2. EGFs
3. MGFs

B. COMPLEX ASYMPTOTICS

4. Rational & Meromorphic
5. Applications of R&M
6. Singularity Analysis
7. Applications of SA
8. Saddle point

http://ac.cs.princeton.edu
The Symbolic Method

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

Unlabelled classes lead to OGFs

Ex. Cartesian product

Construction for ordered pairs,

\[ < A > = < B > \times < C > \]

class of objects \( \alpha = (\beta, \gamma) \) where \( \beta \in < B > \) and \( \gamma \in < C > \)

corresponding counting sequences,

\( A_n, B_n, C_n \)

and ordinary generating functions (OGFs).

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

\[ B(z) = \sum_{\beta \in B} z^{|\beta|} \]

\[ C(z) = \sum_{\gamma \in C} z^{|\gamma|} \]

OGF of Cartesian product is product of OGFs by distributive law

\[ A(z) = \sum_{(\beta, \gamma) \in B \times C} z^{|\beta|+|\gamma|} = \sum_{\beta \in B} z^{|\beta|} \sum_{\gamma \in C} z^{|\gamma|} = B(z) \cdot C(z) \]
The Symbolic method

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

Unlabelled classes lead to OGFs

<table>
<thead>
<tr>
<th>Construction</th>
<th>OGF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Union</td>
<td>( A(z) = B(z) + C(z) )</td>
</tr>
<tr>
<td>Product</td>
<td>( A(z) = B(z) \cdot C(z) )</td>
</tr>
<tr>
<td>Sequence</td>
<td>( A(z) = \frac{1}{1 - B(z)} )</td>
</tr>
<tr>
<td>Powerset</td>
<td>( A(z) = \exp(B(z)) - B(z)^2/2 + B(z)^3/3 + \ldots )</td>
</tr>
<tr>
<td>Multiset</td>
<td>( A(z) = \exp(B(z)) + B(z)^2/2 + B(z)^3/3 + \ldots )</td>
</tr>
<tr>
<td>Cycle</td>
<td>( A(z) = \ln \frac{1}{1 - B(z)} + \frac{1}{2} \ln \frac{1}{1 - B(z)^2} + \ldots )</td>
</tr>
</tbody>
</table>

[ several others ]
The Symbolic Method

Elementary OGF examples (unlabelled objects).

### Nonnegative Integers

- an integer (in unary)
  \[
  1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1
  \]

- atomic class
  \[
  \langle Z \rangle := 1
  \]

- construction for class of all nonnegative integers
  \[
  \langle I_{\geq 0} \rangle = \text{SEQ}(\langle Z \rangle)
  \]

- 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
  \[
  0\ 1\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ 1\ 1\ 1
  \]

- atomic class
  \[
  \langle Z \rangle := 0 \mid 1
  \]

- construction for class of all binary words
  \[
  \langle W \rangle = \text{SEQ}(\langle Z \rangle + \langle Z \rangle)
  \]

- 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
  \[ < Z > := \bullet \]

- construction for class of all binary trees
  \[ < B > := \epsilon + < B > \times < Z > \times < B > \]

- 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
  \[ < Z > := \bullet \]

- construction for class of all general trees
  \[ < G > = < Z > \text{SEQ}(< G >) \]

- 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
\(< Z > := \bullet \)

construction for class of all unary-binary trees
\(< U > = < Z > ( < 1 > + < U > + < U > \times < U > )\)

classical 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**

<table>
<thead>
<tr>
<th>Construction</th>
<th>OGF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Union</td>
<td>$&lt;A&gt; = &lt;B&gt; + &lt;C&gt;$</td>
</tr>
<tr>
<td>Product</td>
<td>$&lt;A&gt; = &lt;B&gt; \star &lt;C&gt;$</td>
</tr>
<tr>
<td>Sequence</td>
<td>$size \ k$</td>
</tr>
<tr>
<td></td>
<td>$any \ size$</td>
</tr>
<tr>
<td>Set</td>
<td>$size \ k$</td>
</tr>
<tr>
<td></td>
<td>$any \ size$</td>
</tr>
<tr>
<td>Cycle</td>
<td>$size \ k$</td>
</tr>
<tr>
<td></td>
<td>$any \ size$</td>
</tr>
</tbody>
</table>

[ several others ]
The Symbolic Method

Representative EGF examples (labelled objects).

**Derangements**

a derangement

\[
\begin{array}{ccccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
3 & 4 & 8 & 2 & 6 & 7 & 5 & 1 \\
\end{array}
\]

alternate (cycle) representation

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

construction for class of all derangements

\[<D> = \text{SET}(\text{CYC}_{>1}(<Z>))\]

automatic derivation of EGF

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

**Surjections**

a surjection (onto mapping)

\[
\begin{array}{ccccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
2 & 1 & 2 & 3 & 5 & 3 & 4 & 3 \\
\end{array}
\]

alternate (preimage) representation

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

construction for class of all surjections onto any initial segment of the integers

\[<R> = \text{SEQ}(\text{SET}_{\geq1}(<Z>))\]

automatic derivation of EGF

\[
R(z) = \frac{1}{1 - \sum_{k\geq1} \frac{z^k}{k!}} = \frac{1}{1 - (e^z - 1)}
= \frac{1}{2 - e^z}
\]
Recovering coefficients from GFs

is sometimes easy, but often challenging

examples

<table>
<thead>
<tr>
<th>class</th>
<th>GF</th>
<th>expansion</th>
</tr>
</thead>
<tbody>
<tr>
<td>binary words</td>
<td>$W(z) = \frac{1}{1 - 2z}$</td>
<td>$1 + 2z + (2z)^2 + (2z)^3 + \ldots$</td>
</tr>
<tr>
<td>trees</td>
<td>$G(z) = \frac{1 + \sqrt{1 - 4z}}{2}$</td>
<td>binomial</td>
</tr>
<tr>
<td>BSTs</td>
<td>$C(z) = \frac{2}{(1 - z)^2} \ln \frac{1}{1 - z}$</td>
<td>elementary convolution</td>
</tr>
<tr>
<td>permutations with all cycle lengths &gt; 3</td>
<td>$D^{(3)}(z) = \frac{e^{z+z^2/2+z^3/3}}{(1 - z)}$</td>
<td>triple convolution</td>
</tr>
<tr>
<td>unary-binary trees</td>
<td>$U(z) = \frac{1 - z - \sqrt{(1 + z)(1 - 3z)}}{2z}$</td>
<td>not elementary</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>singularity type</th>
<th>method of analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>meromorphic (just poles)</td>
<td>Cauchy (elementary)</td>
</tr>
<tr>
<td>fractional powers</td>
<td>Cauchy (Flajolet-Odlyzko)</td>
</tr>
<tr>
<td>logarithmic</td>
<td></td>
</tr>
<tr>
<td>none (entire function)</td>
<td>saddle point</td>
</tr>
</tbody>
</table>

*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 contour

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

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

<table>
<thead>
<tr>
<th>class</th>
<th>GF</th>
<th>singularity type</th>
<th>at</th>
<th>coefficient asymptotics</th>
</tr>
</thead>
<tbody>
<tr>
<td>binary words</td>
<td>$W(z) = \frac{1}{1 - 2z}$</td>
<td>pole</td>
<td>$\frac{1}{2}$</td>
<td>$W_N = 2^N$</td>
</tr>
<tr>
<td>derangements</td>
<td>$D(z) = \frac{e^{-z}}{1 - z}$</td>
<td>pole</td>
<td>1</td>
<td>$D_N \sim e^{-1}$</td>
</tr>
<tr>
<td>surjections</td>
<td>$R(z) = \frac{1}{2 - e^z}$</td>
<td>poles</td>
<td>$\ln 2$</td>
<td>$R_N \sim \frac{N!}{2(\ln 2)^{N+1}}$</td>
</tr>
<tr>
<td>trees</td>
<td>$G(z) = \frac{1 + \sqrt{1 - 4z}}{2}$</td>
<td>square root</td>
<td>$\frac{1}{4}$</td>
<td>$G_N \sim \frac{4^{N-1}}{\sqrt{\pi N^3}}$</td>
</tr>
<tr>
<td>BSTs</td>
<td>$C(z) = \frac{2}{(1 - z)^2} \ln \frac{1}{1 - z}$</td>
<td>logarithmic</td>
<td>1</td>
<td>$C_N \sim 2N \ln N$</td>
</tr>
</tbody>
</table>
“If you can specify it, you can analyze it”

specification $\rightarrow$ Symbolic Method $\rightarrow$ GF $\rightarrow$ Transfer Theorems $\rightarrow$ asymptotics

permutations with all cycle length $> 3$

$$< D^{(3)} > = \text{SET}(\text{CYC}_{>3}(< Z >))$$

$$D^{(3)}(z) = \frac{e^{z+z^2/2+z^3/3}}{(1-z)}$$

$$D^{(3)}_N \sim e^{-1/2-1/3}$$

unary-binary trees

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

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

$$U_N \sim 3^N \sqrt{\frac{3}{4\pi N^3}}$$
AC Schemas

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

Ex. Context-free specifications

Develop a system of combinatorial constructions,

\[
\begin{align*}
\langle G_0 \rangle &= OP_0(\langle G_0 \rangle, \langle G_1 \rangle, \ldots, \langle G_t \rangle) \\
\langle G_1 \rangle &= OP_1(\langle G_0 \rangle, \langle G_1 \rangle, \ldots, \langle G_t \rangle) \\
&\quad \ldots \\
\langle G_t \rangle &= OP_t(\langle G_0 \rangle, \langle G_1 \rangle, \ldots, \langle G_t \rangle)
\end{align*}
\]

which directly maps to a system of GF equations

\[
\begin{align*}
G_0(z) &= F_0(G_0(z), G_1(z), \ldots, G_t(z)) \\
G_1(z) &= F_1(G_0(z), G_1(z), \ldots, G_t(z)) \\
&\quad \ldots \\
G_t(z) &= F_t(G_0(z), G_1(z), \ldots, G_t(z))
\end{align*}
\]

that we can manipulate algebraically to get a single complex function

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

that is amenable to singularity analysis

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

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

Symbolic method leads to a system of implicit function definitions

Groebner basis elimination

Drmota-Lalley-Woods theorem

A universal law for context-free specifications
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
  
  ...
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

Combinatorial Structures

Symbolic Methods
Generating functions (OGFs, EGFs, MGFs)

Complex Asymptotics
Singularity Analysis
Saddle Point

Random Structures
Multivariate Asymptotics
Singularity Perturbation

Asymptotic Counting
Moments of Parameters
Limit Laws
Large Deviations

Exact Counting
Perspective
To analyze an algorithm:

- Implement the algorithm completely.
- Identify unknown quantities representing the basic operations.
- Determine the cost of each basic operation.
- Develop a realistic model for the input.
- Analyze the frequency of execution of the unknown quantities.
- Calculate the total running time (sum of frequency time cost for all quantities).

Benefits:
- Scientific foundation for AofA.
- Can predict performance and compare algorithms.

Challenges:
- Analysis can be difficult and detailed.
- Realistic models can exacerbate the situation.
Theory of Algorithms (AHU, mid-1970s; CLR, present day)

To address Knuth challenges:

• 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.

Benefit: Enabled a new Age of Algorithm Design.

Drawback: Cannot use to predict performance or compare algorithms.

(An elementary fact that is often overlooked!)
Analytic combinatorics

A modern approach to the analysis of algorithms

**Goal:** Accurate and precise analysis, to predict costs (same as Knuth).

To address Knuth challenges:

- **Develop model-building tools.**
  (model is essential, as in any science)

- **Symbolic method + Complexification.**
  (takes detail out of analysis).

**Benefit:** A calculus of discrete structures, of wide applicability.
Dissemination
It’s on the web.

Free .pdf of *Analytic Combinatorics* is available on the web.

Coming soon: New edition of *Introduction to Analysis of Algorithms*

Free .pdf?
As a technology, the book is like a hammer. That is to say, it is perfect: a tool ideally suited to its task. Hammers can be tweaked and varied but will never go obsolete. Even when builders pound nails by the thousand with pneumatic nail guns, every household needs a hammer.

...Now even modest titles have been granted a gift of unlimited longevity. What should an old-fashioned book publisher do with this gift? Forget about cost-cutting and the mass market. Don’t aim for instant blockbuster successes. You won’t win on quick distribution, and you won’t win on price. Cyberspace has that covered.

Go back to an old-fashioned idea: that a book, printed in ink on durable paper, acid-free for longevity, is a thing of beauty.

Make it as well as you can. People want to cherish it.
Sedgewick-Wayne publishing model

Two components
  • traditional textbook (priced to own)
  • forward-looking booksite (free)

Textbook
  • traditional look-and-feel
  • builds on 500 years of experience
  • for use while learning

Booksite
  • supports search
  • has code, test data, animations
  • links to references
  • a living document
  • for use while programming, exploring
Sedgewick-Wayne publishing model

Two components

- traditional textbook (priced to own)
- forward-looking booksite (free)

Textbook

- traditional look-and-feel
- builds on 500 years of experience
- for use while learning

Booksite

- supports search
- has code, test data, animations
- links to references
- a living document
- for use while programming, exploring

Status report: 1.2 million unique visitors in 2011
AofA and AC booksites

in preparation, spring 2012

http://aofa.cs.princeton.edu

http://ac.cs.princeton.edu

Challenge: Provide resources for teaching and learning AofA and AC.
Legacies
PF’s legacy I: Analytic combinatorics

A calculus for the study of discrete structures.

For a brief overview, see

*Analytic Combinatorics—A Calculus of Discrete Structures*
by Philippe Flajolet (*SODA 2007*).
**Recent research in AC**

**Example 1. A Master Theorem for discrete divide-and-conquer recurrences**

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

<table>
<thead>
<tr>
<th>recurrence</th>
<th>solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>(T(n) = n + T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil))</td>
<td>(T(n) \sim n \log n + \Psi(\log n))</td>
</tr>
<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>

**Significance.** Provides *precise results* for use in scientific analysis.

Reference: *A Master Theorem for Discrete Divide-and-Conquer Recurrences*

by Michael Drmota and Wojciech Szpankowski (*SODA 2010*)
Recent research in AC

Example 2. Models for discrete structures in biochemistry

Significance. Symbolic method as a *modeling tool* for scientific applications

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

Recent research in AC

**Example 3.** Random generation of large discrete structures

ternary mapping  
binary tree (Catalan)  
plane partition

series-parallel circuit

**Significance.** Ability to test models and formulate hypotheses in applications

“If you can specify it, you can generate a large one”

**Reference:** Boltzmann Samplers for the Random Generation of Combinatorial Structures
by Philippe Duchon, Philippe Flajolet, Guy Louchard, and Gilles Schaeffer
PF’s legacy II: 200+ coauthors
PF’s legacy III: AofA conferences and events

International Meetings on Probabilistic, Combinatorial, and Asymptotic Methods in the Analysis of Algorithms

1. Dagstuhl (Germany, AofA 1993)
2. Dagstuhl (Germany, AofA 1995)
3. Dagstuhl (Germany, AofA 1997)
5. Barcelona (Spain, AofA 1999)
7. Versailles (France, MathInfo 2000)
8. Tatihou (France, AofA 2001)
10. Versailles (France, MathInfo 2002)
13. Vienna (Austria, MathInfo 2004)
14. Barcelona (Spain, AofA 2005)
15. Alden Biesen (Belgium, AofA 2006)
17. Juan des Pins (France, AofA 2007)
18. Maresias (Brazil, AofA 2008)
19. Blaubeuren (Germany, MathInfo 2008)
20. Frejus (France, AofA 2009)
21. Vienna (Austria, AofA 2010)
22. Będlewo (Poland, AofA 2011)

http://luc.devroye.org/AofA2012.html
PF’s legacy III: AofA conferences and events

AofA 2012
Montreal June 17-23

AofA 2011
Bedlwo June 19-26

PFAC 2011
Paris, December 14-16

ANALCO 2012
Kyoto January 12

LATIN 2012
Arequipa April 17-23

ANALCO 2013
New Orleans January 6

AofA 2013
Menorca May 27-31

CanaDAM 2013
St. John’s June 10-13

AofA 2014
Paris
PF’s legacy IV: Collected Works

to appear, Cambridge University Press.

200+ papers (with 200+ coauthors)
5000+ pages

All papers available online (at PFs website, or ask a friend)

“When you read a paper of Philippe’s, you will always learn something.”

H. K. Hwang
Thank you, Philippe. It is a pleasure to be working with you.

Philippe Flajolet 1948–2011
From Analysis of Algorithms to Analytic Combinatorics

a journey with Philippe Flajolet

Robert Sedgewick
Princeton University