Available online at www.sciencedirect.com

SCIENCE DIRECT®

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

Journal of Computational and Applied Mathematics 174 (2005) 271–313

www.elsevier.com/locate/cam

Singularity analysis, Hadamard products, and tree recurrences

James Allen Fill[a], Philippe Flajolet[b], Nevin Kapur[c,*]

[a]*Department of Applied Mathematics and Statistics, The Johns Hopkins University, 34th and Charles Streets, Baltimore, MD 21218-2682, USA*

[b]*Algorithms Project, INRIA, F-78153 Rocquencourt, France*

[c]*Department of Computer Science, California Institute of Technology, MC 256-80, 1200 E California Blvd., Pasadena, CA 91125, USA*

Received 13 June 2003; received in revised form 29 April 2004

Abstract

We present a toolbox for extracting asymptotic information on the coefficients of combinatorial generating functions. This toolbox notably includes a treatment of the effect of Hadamard products on singularities in the context of the complex Tauberian technique known as singularity analysis. As a consequence, it becomes possible to unify the analysis of a number of divide-and-conquer algorithms, or equivalently random tree models, including several classical methods for sorting, searching, and dynamically managing equivalence relations.

MSC: 05A16 (primary); 40E99; 68W40 (secondary)

Keywords: Singularity analysis; Hadamard products; Tree recurrences; Asymptotic expansions; Generating functions; Divide-and-conquer; Shape functional; Generalized polylogarithm; Contour integration; Moment pumping

This study was motivated by a desire to unify the analysis of a number of algorithms and data structures of computer science. By analysis we mean here (precise) average-case analysis of cost functions as introduced by Knuth and illustrated in the collection [41] as well as in his monumental series, *The Art of Computer Programming* (see especially [39,40]). In the first part of this paper (Sections 1 and 2), we consider a major paradigm of algorithmic design, the “divide-and-conquer” principle, which is closely related to families of random trees and associated “tree recurrences”. The basic framework is described

* Corresponding author. Tel.: +1-626-395-6715.

E-mail addresses: jimfill@jhu.edu (J.A. Fill), philippe.flajolet@inria.fr (P. Flajolet), nkapur@cs.caltech.edu (N. Kapur).

URL: http://www.ams.jhu.edu/∼fill, http://algo.inria.fr/flajolet/, http://www.cs.caltech.edu/∼nkapur.

doi:10.1016/j.cam.2004.04.014

in Section 1, while lead examples are introduced in Section 2. Our treatment rests on combinatorial generating functions.

The central part of this paper (Sections 3 and 4) is devoted to the process of extracting coefficients, at least asymptotically, from generating functions. Singularities have long been recognized to contain highly useful information in this regard, and we start by recalling in Section 3 the basic principles of the complex Tauberian approach known as "*singularity analysis*". Applications to algorithms and trees require, in particular, techniques for coping with generating functions that may be constructed by a tower of several transformations. Here, we develop the theory of composition of singularities under Hadamard products in Section 4. (The reader only interested in complex-analytic aspects can jump directly to Sections 3 and 4.)

The final part (Sections 5 and 6) returns to the original problem of analyzing divide-and-conquer algorithms, taking full advantage of the analytic results of previous sections. Tree recurrences and first moments form the subject of Section 5, where full asymptotic expansions are derived for expectations of costs. Section 6 describes possible extensions of the basic framework to the determination of variances and higher moments as well as to some other random tree models.

1. Introduction

"Divide-and-conquer" is a major principle of algorithmic design in computer science. An instance (I) of a problem to be solved is first split into smaller subproblems (I', I'') that are solved recursively by the same process; the partial solutions are then woven back to yield a solution to the original problem. The abstract scheme is then of the form

$$\begin{aligned} \mathrm{solve}(I) :=\, & (I', I'') := \mathrm{split}(I); \\ & J' := \mathrm{solve}(I');\ J'' := \mathrm{solve}(I''); \\ & \mathrm{return\ weave}(J', J''). \end{aligned} \tag{1}$$

(Problems of size smaller than a certain threshold are treated directly without any recursive call.) Algorithms resorting to scheme (1) include classical sorting methods (mergesort, quicksort, radix-exchange sort), data structures based on trees (binary search trees, digital trees known as "tries", quadtrees for multidimensional search, union–find trees) as well as various methods used in computational geometry, distributed computation, and communication theory. We refer the reader to classical books on data structures, algorithms, and analysis of algorithms for details, for instance, [10,31,35,40,47,48,57,58,60,62].

In general, a class of probabilistic models $\mathfrak{M}_n$ indexed by the size n of the problem instance is assumed to reflect the nature of data fed to the algorithm. A cost function—typically, the number of certain operations performed by the algorithm—then becomes a random variable X_n whose form is induced by $\mathfrak{M}_n$ and the particular divide-and-conquer algorithm considered. The problem is then to obtain characteristics of X_n, for instance its mean, higher moments, or even distributional information. The asymptotic limit $n \to \infty$ is usually considered, since an important phenomenon of "asymptotic simplification" is to be expected in a large number of situations.

Under natural conditions, a recurrence that closely mimics the recursive structure of (1) relates the random variables X_n:

$$X_n = t_n + X_{K_n} + \widetilde{X}_{n-a-K_n}. \tag{2}$$