EXPERIMENTAL METHODS APPLIED TO THE COMPUTATION OF INTEGER SEQUENCES

BY ERIC SAMUEL ROWLAND

A dissertation submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Mathematics

Written under the direction of

Doron Zeilberger

and approved by

PREVIEW

New Brunswick, New Jersey

May, 2009

UMI Number: 3379080

All rights reserved

INFORMATION TO ALL USERS
The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript
and there are missing pages, these will be noted. Also, if material had to be removed,
a note will indicate the deletion.

PREVIEW

UMI 3379080
Copyright 2009 by ProQuest LLC.
All rights reserved. This edition of the work is protected against
unauthorized copying under Title 17, United States Code.

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

ABSTRACT OF THE DISSERTATION

Experimental methods applied to the computation of integer sequences

by Eric Samuel Rowland

Dissertation Director: Doron Zeilberger

PREVIEW

We apply techniques of experimental mathematics to certain problems in number theory and combinatorics. The goal in each case is to understand certain integer sequences, where foremost we are interested in computing a sequence faster than by its definition. Often this means taking a sequence of integers that is defined recursively and rewriting it without recursion as much as possible. The benefits of doing this are twofold. From the view of computational complexity, one obtains an algorithm for computing the system that is faster than the original; from the mathematical view, one obtains new information about the structure of the system.

Two particular topics are studied with the experimental method. The first is the recurrence

$$a(n) = a(n-1) + \gcd(n, a(n-1)),$$

which is shown to generate primes in a certain sense. The second is the enumeration of binary trees avoiding a given pattern and extensions of this problem. In each of these problems, computing sequences quickly is intimately connected to understanding the structure of the objects and being able to prove theorems about them.

Acknowledgements

I greatly thank Doron Zeilberger for his help over the past few years. As an advisor he has been a source of challenging problems and valuable assistance drawn from his comprehensive knowledge of symbolics. As a mathematician he has influenced my philosophy and methodology, and he has provided a practical model for discovering and proving theorems by computer. And as a member of academia he has shown me that rules aren't to be taken too seriously.

I thank the other members of my dissertation committee — Richard Bumby, Stephen Greenfield, and Neil Sloane — for their participation, enthusiasm, and insightful questions.

Thanks are due to an anonymous referee, whose critical comments greatly improved the exposition of the material in Chapter 2.

Regarding the material in Chapter 3, I thank Phillipe Flajolet for helping me understand the relation of the work to existing literature, and I thank Lou Shapiro for suggestions which clarified some points.

I am also indebted to Elizabeth Kupin for much valuable feedback. Her comments greatly improved the exposition and readability of Chapter 3. In addition, the idea of looking for bijections between trees avoiding s and trees avoiding t that do not extend to bijections on the full set of binary trees is hers, and this turned out to be an important generalization of the two-rule bijections I had been considering.

PREVIEW

Table of Contents

PREVIEW

PREVIEW

List of Tables

PREVIEW

List of Figures

PREVIEW

Chapter 1

Introduction

1.1 Overview

This thesis applies techniques of automatic data generation, analysis, form-fitting, and proof to problems in discrete mathematics — in particular to the problem of speeding up the computation of terms in certain integer sequences.

This chapter is devoted to a description of the experimental methodology used throughout the thesis and to the notion of an ansatz.

In Chapter 2 I discuss the recurrence

$$a(n) = a(n-1) + \gcd(n, a(n-1))$$

with initial condition $a(1) = 7$ and prove that $a(n)-a(n-1)$ takes on only 1s and primes, making this recurrence a rare "naturally occurring" generator of primes. Toward a generalization of this result to an arbitrary initial condition, we also study the limiting behavior of $a(n)/n$ and a transience property of the evolution. This work was published in the *Journal of Integer Sequences* [25].

Chapter 3 considers the enumeration of trees avoiding a contiguous pattern. We provide an algorithm for computing the generating function that counts the n-leaf binary trees avoiding a given binary tree pattern t. Equipped with this counting mechanism, we study the analogue of Wilf equivalence in which two tree patterns are equivalent if the respective n-leaf trees that avoid them are equinumerous. We investigate the equivalence classes combinatorially, finding some relationships to Dyck words avoiding a given subword. Toward establishing bijective proofs of tree pattern equivalence, we develop a general method of restructuring trees that conjecturally succeeds to produce an explicit bijection. This work has been submitted for publication [26].

PREVIEW

Results in mathematics are generally presented as static works, where evidence of the dynamic discovery process has been removed. Aside from being the current cultural norm, one reason for this is that conveying the history of a result is difficult and messy. However, I believe that the human process that led to the discovery of a conjecture or theorem or proof is the best way for another human to develop an understanding of that material (short of recreating it from scratch). Therefore, possibly at the expense of some elegance and brevity, I have attempted to indicate how the results in this thesis were found.

The software used for this work was predominantly *Mathematica*. Several packages developed by the author are available from `http://math.rutgers.edu/~erowland/programs.html`. In particular, the package TreePatterns [27] accompanies Chapter 3. Additionally, the algebra software Singular was used via an interface package by Manuel Kauers and Viktor Levandovskyy [18] to compute Gröbner bases for systems of polynomial equations in Chapter 3.

Sequence numbers such as A000108 refer to entries in the Encyclopedia of Integer Sequences [29].

PREVIEW

1.2 Experimental methodology

Experimental mathematics is not a new way of doing mathematics. In fact, it is the oldest way of doing mathematics, the idea being that by naively generating explicit *numeric* examples of a mathematical structure one can eventually perceive the general *symbolic* structure. In practice, the methodology is as follows. Generate data in the form of a sequence of integers or a graph or an image, etc. Then manipulate the data until it takes a recognizable form, e.g., the Thue–Morse sequence or the 4-dimensional hypercube graph or the Sierpiński sieve. Then undo the manipulations symbolically to arrive at an algorithm for computing the original data. Most often this algorithm represents a compression of the data in which its structure is revealed and by which it may be computed more quickly.

For example, if we compute a function $f(n)$ for $n = 1, 2, \ldots, 8$ and find that the values are $1, 1, 2, 3, 5, 8, 13, 21$ then (without requiring much analysis in this case) we generally suspect that $f(n)$ is the nth Fibonacci number.

Although this principle is quite old, what is relatively new is the use of computers to assist in mathematics research. There are major advantages that computers provide in both the stages of data generation and data analysis.

The advantage in generating data is the obvious one — that machines can simply compute faster and more accurately than humans, so any algorithmic task can be done on a much larger scale by machine. Sometimes the time or space needed to compute more data grows quickly, making it infeasible to compute by hand. Sometimes small values of the data do not fit into the general pattern (e.g., the first few terms of an eventually periodic sequence), and the transience may in fact be long. In both these cases it is clearly desirable to compute by machine.

The advantage in using modern mathematical software to study empirical data is that the ease of setting up and executing computations with these systems allows the mathematician to search for structure in real time. One can process the data in many different ways fairly quickly — applying transformations of all sorts, looking for patterns visually in the form of a plot or a graph, attempting fits to known ansatzes (as discussed below), etc. Since the software is doing all the computation, there is little overhead for the human, who is free to experiment with the data as much as it wishes. This naturally increases the likelihood that significant patterns will be discovered.

PREVIEW

1.3 The notion of ansatz

Humans are quite good at identifying some patterns. For example, it is easy for a human (at least a human who has seen them before) to guess a general form for each of the following sequences.

$$1, 4, 9, 16, \dots$$

$$2, 3, 5, 7, 11, 13, 17, 19, \dots$$

$$1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1, 5, \dots$$

However, this process of identification by human familiarity is not very robust. The following sequences are of roughly the same complexity, in a certain mathematical sense, as the preceding sequences, but they are not as immediately identifiable (especially out of context of the preceding sequences).

$$2, 7, 14, 23, \ldots$$

$$3, 7, 13, 19, 29, 37, 43, 53, \ldots$$

$$1, 1, 2, 1, 2, 2, 3, 1, 2, 2, 4, 2, 3, 3, 4, 1, \ldots$$

That is, these sequences are not of the same complexity, in the sense of some humans, as the first sequences. Consequently, we do not want to relegate pattern recognition to humans alone.

Lookup tables such as the Encyclopedia of Integer Sequences [29] and analogous databases for leading digits of real numbers [2, 23] provide one type of systematic pattern recognition. In essence they extend the basic "recognizable primitives".

How can we potentially recognize infinitely many different objects? We might start by applying a finite set of transformations to all of our recognizable primitives. However, no finite list will get us there. (We may of course iterate the transformations, but then in attempting to identify an object we never know when to stop applying the inverse transformations.)

We must work symbolically. To get software to "find a pattern" in empirical data, we specify precisely the general form — the ansatz — of the pattern we are looking for. The ansatz is the symbolic structure behind a class of objects, where each object in the class is realized for certain values of the parameters.

Let us focus on ansatzes of integer sequences, with the understanding that the principles apply equally well to other objects. (For example, in Chapter 3 we consider several ansatzes of bijections on binary trees.) Frequently in discrete mathematics the answer to a question can be rendered as a sequence of integers, and because they are so universal there is much that we know about them.

Some historically successful ansatzes of integer sequences include (in roughly increasing sophistication) periodic functions, polynomials, rational functions, quasi-polynomials,

PREVIEW

C-recursive sequences (solutions of linear recurrences with constant coefficients), k-regular sequences as introduced by Allouche and Shallit [1], sequences whose generating functions are algebraic, and holonomic sequences (solutions of linear recurrences with polynomial coefficients). Zeilberger [33] discusses many of these in greater detail. Each of these ansatzes is useful in sequence identification problems because of its ubiquity in mathematics. The sequences of Chapter 3, for instance, are all algebraic.

Given an ansatz and some data, it is generally routine to find (if it exists) an object in that ansatz that represents the data. If the empirical data can be generated by a function with fewer degrees of freedom than the data, then most likely the sequence has been identified. A familiar example is the interpolation of polynomials: If a polynomial of degree 2 correctly reproduces 4 terms of a sequence, this indicates some redundancy in those terms.

Certainly one comes across objects in mathematics that do not fit a known ansatz. When this happens it is the role of the human to study the object until its structure becomes clear. That is, the human is the creator/identifier of new ansatzes, and this is the not-yet-routine mathematics being done in the context of the experimental method. Such was the case with the integer sequences in Chapter 2 and with the bijections in Chapter 3; these new classes were introduced to answer particular number theoretic and combinatorial questions.

PREVIEW

Chapter 2

A natural prime-generating recurrence

2.1 Introduction

Since antiquity it has been intuited that the distribution of primes among the natural numbers is in many ways random. For this reason, functions that reliably generate primes have been revered for their apparent traction on the set of primes.

Ribenboim [24, page 179] provides three classes into which certain prime-generating functions fall:

(a) $f(n)$ is the nth prime p_n.

(b) $f(n)$ is always prime, and $f(n) \neq f(m)$ for $n \neq m$.

(c) The set of positive values of f is equal to the set of prime numbers.

Known functions in these classes are generally infeasible to compute in practice. For example, both Gandhi's formula

$$p_n = \left\lfloor 1 - \log_2 \left(-\frac{1}{2} + \sum_{d \mid P_{n-1}} \frac{\mu(d)}{2^d - 1} \right) \right\rfloor$$

[11], where $P_n = p_1 p_2 \cdots p_n$, and Willans' formula

$$p_n = 1 + \sum_{i=1}^{2^n} \left\lfloor \left(\frac{n}{\sum_{j=1}^{i} \left\lfloor \left(\cos \frac{(j-1)!+1}{j} \pi \right)^2 \right\rfloor} \right)^{1/n} \right\rfloor$$

[31] satisfy condition (a) but are essentially versions of the sieve of Eratosthenes [12, 13]. Gandhi's formula depends on properties of the Möbius function $\mu(d)$, while Willans' formula is built on Wilson's theorem. Jones [16] provided another formula for p_n using Wilson's theorem.

PREVIEW