BOLTZMANN SAMPLERS FOR THE RANDOM GENERATION OF COMBINATORIAL STRUCTURES

PHILIPPE DUCHON, PHILIPPE FLAJOLET, GUY LOUCHARD, GILLES SCHAEFFER

Abstract. This article proposes a surprisingly simple framework for the random generation of combinatorial configurations based on what we call *Boltzmann models*. The idea is to perform random generation of possibly complex structured objects by placing an appropriate measure spread over the whole of a combinatorial class—an object receives a probability essentially proportional to an exponential of its size. As demonstrated here, the resulting algorithms based on real-arithmetic operations often operate in linear time. They can be implemented easily, be analysed mathematically with great precision, and, when suitably tuned, tend to be very efficient in practice.

1. Introduction

In this study, *Boltzmann models* are introduced as a framework for the random generation of structured combinatorial configurations, like words, trees, permutations, constrained graphs, and so on. A Boltzmann model relative to a combinatorial class $\mathcal{C}$ depends on a *real-valued* (continuous) control parameter $x > 0$ and places an appropriate measure that is spread over the whole of $\mathcal{C}$: This measure is essentially proportional to $x^{|\omega|}$ for an object $\omega \in \mathcal{C}$ of size $|\omega|$. Random objects under a Boltzmann model then have a fluctuating size, but objects with the same size invariably occur with the same probability. In particular, a *Boltzmann sampler* (i.e., a random generator that produces objects distributed according to a Boltzmann model) draws *uniformly* at random an object of size n, when the size of its output is conditioned to be the fixed value n.

As we demonstrate, Boltzmann samplers can be derived systematically (and simply) for classes that are specified in terms of a basic collection of general-purpose combinatorial constructions. These constructions are precisely the ones that surface recurrently in modern theories of combinatorial analysis [4, 28, 30, 60, 61] and in systematic approaches to random generation of combinatorial structures [29, 51]. As a consequence, one obtains with surprising ease Boltzmann samplers covering an extremely wide range of combinatorial types.

In most of the combinatorial literature so far, fixed-size generation has been the standard paradigm for the random generation of combinatorial structures, and a vast literature exists on the subject. There, either specific bijections are exploited or general combinatorial decompositions are put to use in order to generate objects at random based on counting possibilities—the latter approach has come to be known as the "recursive method" originating with Nijenhuis and Wilf [51], then systematized and extended by Flajolet, Zimmermann, and Van Cutsem in [29].

Date: Version of January 1,2003. Submitted to *Combinatorics, Probability, and Computing.*

In contrast, the basic principle of Boltzmann sampling is to *relax* the constraint of generating objects of a strictly fixed size, and prefer to draw objects with a randomly varying size. As we shall see, normally, one can then *tune* the value of the control parameter x in order to favour objects of a size in the vicinity of a target value n. (A "tolerance" of, say, a few percents on size of the object produced is likely to cater for many practical simulation needs.) If the tuning mentioned above is not sufficient, one can always pile up a rejection method to restrict further the size of the element drawn. In this way, Boltzmann samplers may be employed for approximate-size as well as fixed-size random generation.

We propose Boltzmann samplers as an attractive alternative to standard combinatorial generators based on the recursive method and implemented in packages like Combstruct (under the computer algebra system MAPLE) and CS (under MUPAD). The algorithms underlying the recursive necessitate a preprocessing phase where tables of integer constants are set up, then they appeal to a boustrophedonic strategy in order to draw a random object of size n. In the abstract, the *integer-arithmetic* complexities attached to the recursive method and measured by the number of (large) *integer-arithmetic* operations are as follows:

(1)

Preproc. memory	*Preproc. time*	*Time per generation*
$O(n)$	$O(n^2)$ or $O(n^{1+\varepsilon})$	$O(n \log n)$

The integer-based algorithms require the costly maintenance of large tables of constants (in number $O(n)$). In fact, they effect arithmetic operations over large multiprecision integers, which themselves have size $O(n)$ (in the unlabelled case) or $O(n \log n)$ (in the labelled case); see [29]. Consequently, the overall Boolean complexities involve an extra factor of $O(n)$ at least, leading to a cost measured in elementary operations that is quadratic or worse. (The integer-arithmetic time of the preprocessing phase could in principle be decreased from $O(n^2)$ to $O(n^{1+\varepsilon})$ thanks to the recent work of van der Hoeven [65], but this does not affect our basic conclusions.) An alternative, initiated by Denise, Dutour, and Zimmermann [12, 13], consists in treating integers as real numbers and approximating them using real arithmetics ("floating-point" implementations), possibly supplementing the technique by adaptive precision routines. In the case of real-based algorithms, the Boolean as well as practical complexities improve, and they become fairly well represented by the data of Equation (1), but the memory and time of the preprocessing phase remains fairly large, while the time per generation remains inherently superlinear.

As we propose to show, Boltzmann algorithms can well be competitive when compared to combinatorial methods: Boltzmann samplers only necessitate a small *fixed* number of low precision real constants that are normally easy to compute while their complexity is always linear in the size of the object drawn. Accordingly, uniform random generation of objects with sizes in the range of millions is becoming a possibility, whenever the Boltzmann framework is applicable. The price to be paid is an occasional loss of certainty in the exact size of the object generated, typically, a *tolerance* on sizes of a few percents should be granted; refer to Figure 10 in the concluding section. The table that summarizes the complexities of Boltzmann generators, measured in *real-arithmetic* operations is then:

(2)

Preproc. memory	*Preproc. time*	*Time per generation*
$O(1)$	"small"	with tolerance : $O(n)$

The vague qualifier "small" refers to the fact that practical implementations will be based on floating point approximations to exact real number arithmetics, in which case, typically, the preprocessing time is likely to be a small power of $\log n$. (That this preprocessing is practically feasible and of a very low complexity should at least transpire from the various examples given, but a systematic discussion would carry us too far away from our main objectives[1].)

As regards random generation, the ideas presented here draw their origins from many sources. First the recursive method of [29, 51] served as a key conceptual guide for delineating the types of objects that are systematically amenable to Boltzmann sampling. Ideas from a statistical physics point of view on combinatorics, of which great use was made by Vershik and his collaborators [10, 67], then provided crucial insight regarding the new class of algorithms for random generation that is presented here. Another important ingredient is the collection of rejection algorithms developed by Duchon, Louchard, and Schaeffer for certain types of trees, polyominos, and planar maps [17, 45, 56]. There are also similarities with the technique of "shifting the mean" (see Greene and Knuth's book [33, p. 78–80]) as well as the theory of large deviations [11] and "exponential families" of probability theory—we have benefited from discussions with Alain Denise on these aspects. Finally, the principles of analytic combinatorics (see [28]) provide essential clues for deciding situations in which the algorithms are likely to be efficient. Further connections are discussed at the end of the next section.

Plan of this study. Boltzmann models and samplers are introduced in Section 2. Boltzmann models exist in two varieties: the ordinary and the exponential models. Ordinary models serve for combinatorial classes that are "unlabelled", the corresponding samplers being developed in Section 3, where basic construction rules are described. Section 4 proceeds in a parallel way with exponential models and "labelled" classes. Some of the complexity issues raised by Boltzmann sampling are examined in Section 5. There it is shown that, at least in the idealized sense of *exact real-number computations*, a Boltzmann sampler suitably equipped with a fixed (and small) number of driving constants operates in time that is *linear* in the (fluctuating) size of the object it produces.

Sections 2 to 5 develop Boltzmann samplers that operate *freely* under the sole effect of the defining parameter x. We examine next the way the control parameter x can be *tuned* to attain objects at or near a target value: this is the subject of Section 6, where rejection is introduced and a technique based on the pointing transformation is developed. Section 7 describes two types of situation where the basic Boltzmann samplers turn out to be *optimized* by assigning a critical value to the control parameter x. Section 8 offers a few concluding remarks.

An extended abstract summarizing several of the results described here has been presented at the ICALP'2002 Conference in Malaga [18].

2. Boltzmann models and samplers

We consider a class $\mathcal{C}$ of combinatorial objects of sorts, with $|\cdot|$ the size function mapping $\mathcal{C}$ to $\mathbb{Z}_{\geq 0}$. By $\mathcal{C}_n$ is meant the subclass of $\mathcal{C}$ comprising all the objects in $\mathcal{C}$ having size n, and each $\mathcal{C}_n$ is assumed to be finite. One may think of binary

[1]The primary goal of this article is on practical algorithmic design, *not* analysis of algorithms, although a fair amount of analysis, by necessity, enters into the discussion.

words (with size defined as length), permutations, graphs and trees of various types (with size defined as number of vertices), and so on. Any set $\mathcal{C}$ endowed with a size function and satisfying the finiteness axiom will henceforth be called a *combinatorial class.*

The *uniform probability distribution* over $\mathcal{C}_n$ assigns to each $\gamma \in \mathcal{C}_n$ the probability

$$\mathbb{P}_{\mathcal{C}_n}\{\gamma\} = 1/C_n,$$

with $C_n := \operatorname{card}(\mathcal{C}_n)$. *Exact-size* random generation means the process of drawing **uniformly** at random from the class $\mathcal{C}_n$. We also consider (see Sections 6 and 7 for a description of various strategies) random generation from "neighbouring classes", $\mathcal{C}_N$ where N may not be totally under control, but should still be in the vicinity of n, namely, in some interval $(1-\varepsilon)n \le N \le (1+\varepsilon)n$, for some "*tolerance*" factor $\varepsilon > 0$; this is called *approximate-size* (uniform) random generation. It must be stressed that, even under approximate-size random generation, *two objects of the same size are invariably drawn with the same probability.*

Definition 1. *The* Boltzmann models *of parameter x exist in two varieties, the ordinary version and the exponential version. They assign to any object $\gamma \in \mathcal{C}$ the following probability:*

$$\textit{Ordinary/Unlabelled case:} \qquad \mathbb{P}_x(\gamma) = \frac{1}{C(x)} \cdot x^{|\gamma|} \quad \textit{with} \quad C(x) = \sum_{\gamma \in \mathcal{C}} x^{|\gamma|},$$

$$\textit{Exponential/Labelled case:} \qquad \mathbb{P}_x(\gamma) = \frac{1}{\widehat{C}(x)} \cdot \frac{x^{|\gamma|}}{|\gamma|!} \quad \textit{with} \quad \widehat{C}(x) = \sum_{\gamma \in \mathcal{C}} \frac{x^{|\gamma|}}{|\gamma|!}.$$

A Boltzmann sampler (or generator) $\Gamma C(x)$ *for a class $\mathcal{C}$ is a process that produces objects from C according to the corresponding Boltzmann model, either ordinary or exponential.*

The normalization coefficients are nothing but the values at x of the counting generating functions, respectively of ordinary type (OGF) for C and exponential type (EGF) for $\widehat{C}$:

$$C(z) = \sum_{n \ge 0} C_n z^n, \qquad \widehat{C}(z) = \sum_{n \ge 0} C_n \frac{z^n}{n!}.$$

Coherent values of x defined to be such that $0 < x < \rho_C$ (or $\rho_{\widehat{C}}$), with ρ_f the radius of convergence of f are to be considered. The quantity ρ_f is referred to as the "critical" or "singular" value. (In the particular case when the generating function $C(x)$ still converges at ρ_C, one may also use the limit value $x = \rho_C$ to define a valid Boltzmann model; see Section 7 for uses of this technique.)

For reasons which will become apparent, we have introduced two categories of models, the ordinary and exponential ones. Exponential Boltzmann models are appropriate for handling *labelled* combinatorial structures while ordinary models correspond to *unlabelled* structures of combinatorial theory[2]. In the unlabelled universe, all elementary components of objects ("atoms") are indistinguishable, while in the labelled universe, they are all distinguished from one another by bearing a distinctive mark, say one of the integers between 1 and n if the object considered

[2]This terminology is standard in combinatorial enumeration and graph theory; see, e.g., the books of Bergeron et al., Goulden–Jackson, Harary–Palmer, Stanley, and Wilf [4, 30, 34, 60, 61, 69] or the preprints by Flajolet & Sedgewick [28].

has size n. Permutations written as sequences of distinct integers are typical labelled objects while words over a binary alphabet appear as typical unlabelled objects made of "anonymous" letters, say $\{a, b\}$ for a binary alphabet.

For instance, consider the (unlabelled) class $\mathcal{W}$ of all binary words, $\mathcal{W} = \{a, b\}^\star$. There are $W_n = 2^n$ words of length n and the OGF is $W(z) = (1-2z)^{-1}$. The probability assigned by the ordinary Botzmann model to any word w is $x^{|w|}(1-2x)$. There, the coherent values of x are all the positive values less than the critical value $\rho_W = \frac{1}{2}$. The probability that a word of length n is selected is $(2x)^n(1-2x)$, so that the Boltzmann model of binary words is logically equivalent to the following process: draw a random variable N according to the geometric distribution of parameter $2x$; if the value $N = n$ is obtained, draw uniformly at random any of the possible words of size n. For the labelled case, consider the class K of all cyclic permutations, $\mathcal{K} = \{[1], [1\,2], [1\,2\,3], [1, 3, 2], \ldots\}$. There are $K_n = (n-1)!$ cyclic permutations of size n over the canonical set of "labels" $\{1, \ldots, n\}$. The EGF is

$$\widehat{K}(z) = \sum_{n\geq 1}(n-1)!\frac{z^n}{n!} = \sum_{n\geq 1}\frac{z^n}{n} = \log\frac{1}{1-z}. \tag{3}$$

The probability of drawing a cyclic permutation of some fixed size n is then,

$$\frac{1}{\log(1-x)^{-1}}\frac{x^n}{n}, \tag{4}$$

a quantity defined for $0 < x < \rho_{\widehat{K}} = 1$. (This is known as the "logarithmic series distribution"; see Section 4). Like in the case of binary words, the Boltzmann model can thus be realized by first selecting size according to the logarithmic series distribution, and then by drawing uniformly at random a cyclic permutation of the chosen size. We are precisely going to *revert* this process and show that, in many cases, it is of advantage to draw *directly* from a Boltzmann model, (Sections 3 to 5), and from there derive random generators that are efficient for a given range of sizes (Sections 6 and 7).

The size of the resulting object under a Boltzmann model is a random variable denoted throughout by N. By construction, the probability of drawing an object of size n is, under the model of index x,

$$\mathbb{P}_x(N = n) = \frac{C_n x^n}{C(x)}, \quad \text{or} \quad \mathbb{P}_x(N = n) = \frac{C_n x^n}{n!\widehat{C}(x)}, \tag{5}$$

for the ordinary and exponential model, respectively. The law is well quantified by the following lemma. (See, e.g., Huang's book [37] for similar calculations from the statistical mechanics angle.)

Proposition 1. *The random size of the object produced under the ordinary Boltzmann model of parameter x has first and second moments satisfying*

$$\mathbb{E}_x(N) = x\frac{C'(x)}{C(x)}, \qquad \mathbb{E}_x(N^2) = \frac{x^2C''(x) + xC'(x)}{C(x)}. \tag{6}$$

The same expressions are valid, but with $\widehat{C}$ replacing C, in the case of the exponential Boltzmann model. In both cases, the expected size $\mathbb{E}_x(N)$ is an increasing function of x.

Proof. Under the ordinary Boltzmann model, the probability generating function of N is

$$\sum_n \mathbb{P}_x(N=n)z^n = \frac{C(xz)}{C(x)},$$

by virtue of (5). The result then immediately follows by differentiation upon setting $z=1$:

$$\mathbb{E}_x(N) = \left(\frac{\partial}{\partial z}\frac{C(xz)}{C(x)}\right)_{z=1}, \qquad \mathbb{E}_x(N(N-1)) = \left(\frac{\partial^2}{\partial z^2}\frac{C(xz)}{C(x)}\right)_{z=1}.$$

The very same calculation applies to exponential Boltzmann models, but with the EGF $\widehat{C}$ then replacing the OGF C.

The mean size $\mathbb{E}_x(N)$ is always a strictly increasing function of x as soon as the class $\mathcal{C}$ contains at least two elements of different sizes. Indeed one verifies by a trite calculation the identity

$$x\frac{d}{dx}\mathbb{E}_x(N) = \mathbb{V}_x(N),$$

where $\mathbb{V}$ denote the variance operator. Since the variance of a nondegenerate random variable is always strictly positive the derivative of $\mathbb{E}_x(N)$ is positive and $\mathbb{E}_x(N)$ is increasing. (This property is in fact a special case of Hadamard's convexity theorem.) ∎

For instance, in the case of binary words, the coherent choice $x=0.4$ leads to a size with mean value 4 and standard deviation about 4.47; for $x=0.49505$, the mean and standard deviation of size become respectively 100 and 100.5. For cyclic permutations, we determine similarly that the choice $x=0.99846$ leads to an object of mean size equal to 100, while the standard deviation is somewhat higher than for words, being equal to 234. In general, the distribution of random sizes under a Boltzmann model, as given by Formula (5), strongly depends on the family under consideration. Figure 1 illustrates three widely differing profiles: for set partitions, the distribution is "bumpy", so that a choice of the appropriate x will most likely generate an object close to the desired size; for surjections (whose behaviour is analogous to the one of binary words), the distribution becomes fairly "flat" as x nears the critical value; for trees, it is "peaked" at the origin, so that very small objects are generated with high probability. It is precisely the purpose of later sections (Sections 6 and 7) to recognize and exploit the "physics" of these distributions in order to deduce efficient samplers for exact and approximate size random generation.

Relation to other fields. The term "Boltzmann model" comes from the great statistical physicist Ludwig Boltzmann whose works (together with those of Gibbs and Maxwell) led to enunciate the following principle: *Statistical mechanical configurations of energy equal to E in a system have a probability*[3] *of occurrence proportional to* $e^{-\beta E}$. If one identifies size of a combinatorial configuration with energy of a thermodynamical system and sets $x = e^{-\beta}$, then what we term the ordinary Boltzmann models become the usual model of statistical mechanics. The counting generating function in the combinatorial world then coincides with the normalization constant in the statistical mechanics world where it is known as the *partition*

[3]Distributions of the type $e^{-\beta E}$ play an important rôle in the study of point processes and they tend to be known to probabilists under the name of "Gibbs measures".

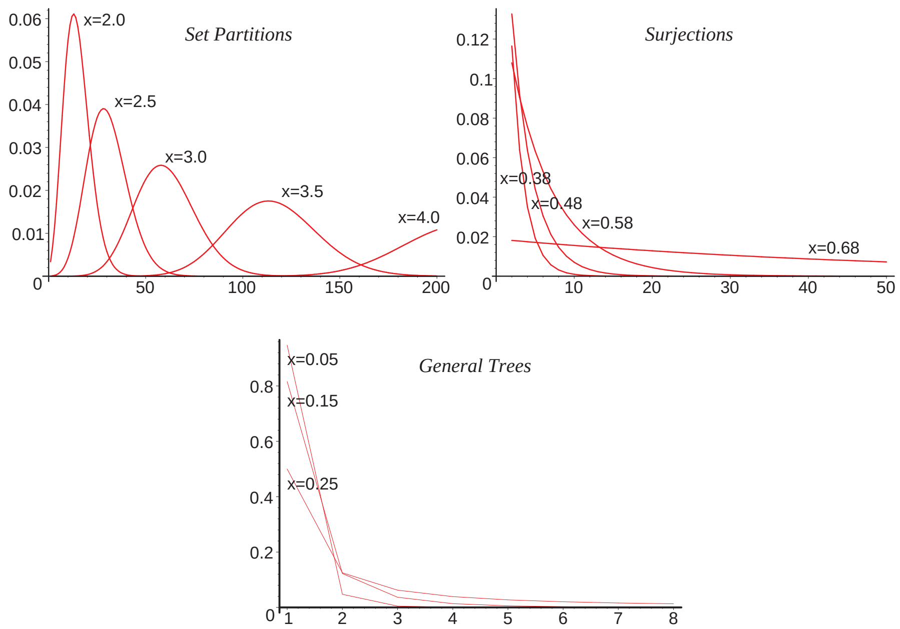


FIGURE 1. Size distributions under Boltzmann models for various values of parameter x. From top to bottom: the "bumpy" type of set partitions (Example 5), the "flat" type of surjections (Example 6), and the "peaked" type of general trees. (Example 2).

function—the *Zustandsumme* often denoted by Z. (Note: In statistical mechanics, $\beta = 1/(kT)$ is an inverse temperature. Thus situations where $x \to 0$ formally correspond to low temperatures or "freezing" and give more weight to small structures, while $x \to \rho^-$ corresponds to high temperatures or "melting", that is, to larger sizes of the combinatorial configurations being generated.)

Exponential weights of the Boltzmann type are naturally essential to the simulated annealing approach to combinatorial optimization. In the latter area, for instance, Fill and Huber [22] have shown the possibility of drawing at random independent sets of graphs according to a Boltzmann distribution, at least for certain values of the control parameter $x = e^{-\beta}$. Closer to us, Compton [7, 8] has made an implicit use of what we call Boltzmann models for the analysis of 0–1 laws and limit laws in logic; see also the account by Burris [6]. Vershik has initiated in a series of papers (see [67] and references therein) a programme that can be described in our terms as first developing the probabilistic study of combinatorial objects under a Boltzmann model and then "returning" to fixed size statistics by means of Tauberian arguments of sorts. (A similar description can be applied to Compton's approach; see especially the work [50] for recent developments in this direction.) As these examples indicate, the general idea of Boltzmann models is certainly not new, and, in this work, we may at best claim originality for aspects related to the fast random generation of combinatorial structures.

Construction		*Generator*
Singleton	$\mathcal{C} = \{\omega\}$	$\Gamma C(x) = \omega$
Union	$\mathcal{C} = \mathcal{A} + \mathcal{B}$	$\Gamma C(x) = \left(\text{Bern}\left(\frac{A(x)}{A(x)+B(x)}\right) \longrightarrow \Gamma A(x) \mid \Gamma B(x)\right)$
Product	$\mathcal{C} = \mathcal{A} \times \mathcal{B}$	$\Gamma C(x) = \big(\Gamma A(x); \Gamma B(x)\big)$
Sequence	$\mathcal{C} = \mathfrak{S}(\mathcal{A})$	$\Gamma C(x) = \big(\text{Geom}(A(x)) \Longrightarrow \Gamma A(x)\big)$

FIGURE 2. The inductive rules for ordinary Boltzmann samplers.

3. Ordinary Boltzmann Generators

In this section and the next one, we develop a collection of rules by which one can assemble Boltzmann generators from simpler ones. The combinatorial classes considered are built by means of a small set of constructions that have wide expressive power. The language in which classes are specified is in essence the same as the one underlying the recursive method [29]: it includes the constructions of union, product, sequence, and, in the labelled case treated in the next section, the additional set and cycle constructions. For each allowable class, a Boltzmann sampler can be derived in an entirely systematic (and even automatic) manner.

A *combinatorial construction* builds a new class $\mathcal{C}$ from structurally simpler classes $\mathcal{A}, \mathcal{B}$, in such a way that $\mathcal{C}_n$ is determined from smaller objects, that is, from elements of $\{\mathcal{A}_j\}_{j=0}^n, \{\mathcal{B}_j\}_{j=0}^n$. The unlabelled constructions considered here are disjoint *union* $(+)$, cartesian *product* $(\times)$, and *sequence* formation $(\mathfrak{S})$. We define these in turn and concurrently build the corresponding Boltzmann sampler ΓC for the composite class $\mathcal{C}$, given random generators $\Gamma A, \Gamma B$ for the ingredients and assuming the values of intervening generating functions $A(x), B(x)$ at x to be real numbers which are *known exactly.*

Finite Sets. Clearly if $\mathcal{C}$ is finite (and in practice small), one can generate a random element of $\mathcal{C}$ by selecting it according to the finite probability distribution defined by the Boltzmann model: If $\mathcal{F} = \{\omega_1, \ldots, \omega_r\}$, then one selects f_j with probability proportional to $z^{|f_j|}$. Thus, drawing from a finite set is equivalent to a finite probabilistic switch. Drawing from a singleton set is then a deterministic procedure which directly outputs the object in question. In particular, in what follows, we make use of the singleton classes, $\mathbf{1}$ and $\mathcal{Z}$, formed respectively of one element of size 0 (analogous to the empty word of formal language theory) and of one element of size 1 that can be viewed as a generic "atom" out of which complex combinatorial structures are formed.

Disjoint union. Write $\mathcal{C} = \mathcal{A} + \mathcal{B}$ if $\mathcal{C}$ is the union of disjoint copies of $\mathcal{A}$ and $\mathcal{B}$, with size on $\mathcal{C}$ inherited from $\mathcal{A}, \mathcal{B}$. By disjointness, one has $C_n = A_n + B_n$, so that

$$C(z) = A(z) + B(z). \tag{7}$$

Consider a random element of $\mathcal{C}$ under the Boltzmann model of index x. Then, the probability that this random element is some $\alpha \in \mathcal{A}$ is

$$\mathbb{P}_{\mathcal{C},x}(\alpha) \equiv \frac{x^{|\alpha|}}{C(x)} = \frac{x^{|\alpha|}}{A(x)} \cdot \left(\frac{A(x)}{C(x)}\right).$$

The Boltzmann model corresponding to $C(x)$ is then a mixture of the models associated to $A(x)$ and $B(x)$, the probability of selecting a particular γ in $\mathcal{C}$ being

$$\mathbb{P}_{\mathcal{C},x}(\gamma \in \mathcal{A}) = \frac{A(x)}{C(x)}, \qquad \mathbb{P}_{\mathcal{C},x}(\gamma \in \mathcal{B}) = \frac{A(x)}{C(x)}.$$

Given a generator for a Bernoulli variable $\mathrm{Bern}(p)$ defined by

$$\mathrm{Bern}(p) = 1 \text{ with probability } p; \quad \mathrm{Bern}(p) = 0 \text{ with probability } 1-p,$$

two Boltzmann samplers $\Gamma A(x), \Gamma B(x)$, and the values of the OGFs $A(x), B(x)$, a Boltzmann sampler ΓC for class $\mathcal{C} = \mathcal{A} + \mathcal{B}$ is simply obtained by the procedure:

function $\Gamma C(x$: real); $\{$generates $\mathcal{C} = \mathcal{A} + \mathcal{B}\}$
let $p_A := A(x)/(A(x) + B(x))$;
if $\mathrm{Bern}(p_A)$ then $\mathrm{return}(\Gamma A(x))$ else $\mathrm{return}(\Gamma B(x))$ fi; end.

We abbreviate this construction as

$$\left(\mathrm{Bern}\left(\frac{A(x)}{C(x)}\right) \longrightarrow \Gamma A(x) \mid \Gamma B(x)\right), \tag{8}$$

where $(X \longrightarrow f \mid g)$ is a shorthand notation for: "if the random variable X is 1, then execute f, else execute g". More generally, if X ranges over a finite set with r elements endowed with a probability measure, $p_1, \ldots, p_r$, we shall use the extended notation

$$(\mathrm{Bern}(p_1, \ldots, p_{r-1}) \longrightarrow f_1 \mid \ldots \mid f_r) \tag{9}$$

to represent the corresponding r-fold probabilistic switch.

Cartesian Product. Write $\mathcal{C} = \mathcal{A} \times \mathcal{B}$ if $\mathcal{C}$ is the set of ordered pairs from $\mathcal{A}$ and $\mathcal{B}$, and size on $\mathcal{C}$ is inherited additively from $\mathcal{A}, \mathcal{B}$. Generating functions satisfy

$$C(z) = A(z) \cdot B(z) \quad \text{since} \quad C(z) = \sum_{\langle \alpha,\beta\rangle \in \mathcal{A}\times\mathcal{B}} z^{|\alpha|+|\beta|}. \tag{10}$$

A random element of $\gamma \in \mathcal{C}$ with $\gamma = (\alpha, \beta)$ then has probability

$$\mathbb{P}_{\mathcal{C},x}(\gamma) \equiv \frac{x^{|\gamma|}}{C(x)} = \frac{x^{|\alpha|}}{A(x)} \cdot \frac{x^{|\beta|}}{B(x)}.$$

It is thus obtained by forming a pair $\langle \alpha, \beta \rangle$ with α, β drawn *independently* from the Boltzmann models $\Gamma A(x), \Gamma B(x)$:

function $\Gamma C(x$: real); $\{$generates $\mathcal{C} = \mathcal{A} \times \mathcal{B}\}$
$\mathrm{return}(\langle \Gamma A(x), \Gamma B(x) \rangle)$ $\{$independent calls$\}$.

We shall abbreviate this schema as

$$\Gamma C(x) = \big(\Gamma A(x); \Gamma B(x)\big),$$

which can be read either as functionally producing a pair, or as sequential execution of the two procedures. We shall also use the natural extension $(f_1; \ldots; f_r)$ when r-tuples are involved.

Sequences. Write $\mathcal{C} = \mathfrak{S}(\mathcal{A})$ if $\mathcal{C}$ is composed of all the finite sequences of elements of $\mathcal{A}$ (with size of a sequence additively inherited from sizes of components). The sequence class $\mathcal{C}$ is also the solution to the symbolic equation $\mathcal{C} = \mathbf{1} + \mathcal{A} \times \mathcal{C}$

(with $\mathbf{1}$ the empty sequence), which only involves unions and products and is reflected by the relation between OGFs: $C = 1 + AC$. Consequently,

$$C(z) = \frac{1}{1 - A(z)}. \tag{11}$$

This gives rise to two logically equivalent designs for a ΓC sampler:

(i) the recursive sampler,

function $\Gamma C(x : \mathsf{real})$; $\{$generates $\mathcal{C} = \mathfrak{S}(\mathcal{A})\}$
if $\mathrm{Bern}(A(x))$ then $\mathrm{return}(\Gamma A(x), \Gamma C(x))$ $\{$recursive call$\}$
else $\mathrm{return}(\mathbf{1})$.

(ii) the geometric sampler,

function $\Gamma C(x : \mathsf{real})$; $\{$generates $\mathcal{C} = \mathfrak{S}(\mathcal{A})\}$
draw k according to $\mathrm{Geom}(A(x))$;
return the k-tuple $\langle \Gamma A(x), \ldots, \Gamma A(x) \rangle$ $\{k$ independent calls$\}$.

The recursive sampler for sequences is built from first principles (union and product rules). It might in principle loop for ever. However, by design, it repeatedly draws a Bernoulli random variable till the value 0 is attained. Thus, the number of components generated is a geometric random variable with rate $A(x)$, where, we recall, X is geometric of rate λ if

$$\mathbb{P}(X = k) = (1 - \lambda)\lambda^k.$$

For coherence to be satisfied, we must have $A(x) < 1$. Then, the recursive sampler halts with probability 1 since the expected number of recursive calls is finite and equal to $(1 - A(x))^{-1}$. This discussion justifies the geometric generator, which unwinds the recursion of the basic recursive sampler using a generator $\mathrm{Geom}(\lambda)$ for the geometric variable of parameter λ.

In what follows, we use the notation,

$$\big(Y \Longrightarrow f\big) \tag{12}$$

to mean: the random variable Y is drawn; if the value $Y = y$ is returned, then y *independent calls*, $f_1, \ldots, f_y$ are launched. The scheme giving the sequence sampler for $\mathcal{C} = \mathfrak{S}(\mathcal{A})$ is then simply:

$$\Gamma C(x) = \left(\mathrm{Geom}(A(x)) \Longrightarrow \Gamma(x)\right).$$

Recursive classes. As suggested by the sequence construction, recursively defined classes admit generators that call themselves recursively. A specification by means of constructors is "well-founded" if it builds objects from smaller ones. An equivalent condition, when no recursion is involved, is that the sequence (and, for exponential Boltzmann models below, set, and cycle) operations are never applied to classes that contain objects of size 0. For recursive structures this is a testable property akin to "properness" in the theory of context-free grammars. (A context-free grammar is proper if the empty word is not generated with infinite multiplicity.) This well-foundedness condition also guarantees that the equations defining generating function equations are well-posed and contracting in the space of formal power series endowed with the standard metric, $\mathrm{dist}(f, g) = 2^{-\mathrm{val}(f-g)}$; accordingly, iteration provides a geometrically converging approximation scheme that makes it possible to determine generating function values for all coherent values of x (by analyticity and dominated convergence). See [27, 29] for a detailed discussion of this topic and the corresponding decision procedures.

Theorem 1. *Define as* specifiable *an unlabelled class that can be* finitely *specified (in a possibly recursive way) from finite sets by means of disjoint unions, cartesian products, and the sequence construction. Let C be an unlabelled specifiable class and x be a coherent parameter in $(0, \rho_C)$. Assume as given an oracle that provides the finite collection of* exact *values at a coherent value x of the generating functions intervening in a specification of a class $\mathcal{C}$. Then, the Boltzmann generator $\Gamma C(x)$ assembled from the definition of $\mathcal{C}$ by means of the four rules summarized in Figure 2 has a complexity measured in the number of $(+, -, \times, \div)$ real-arithmetic operations that is* linear *in the size of its output object.*

Proof. For a coherent value of size, the expectation of size is finite, so that, in particular, size is finite with probability 1. Given a specification Σ for $\mathcal{C}$, each object ω admits a unique parse tree (or syntax tree) $\tau[\omega]$ relative to Σ. For well-founded specifications, this parse tree τ is of a size linear in the size of the object produced. We shall see later (Lemma 1 in Section 5) that in the real-arithmetic model a Bernoulli choice can be effected with complexity $O(1)$ and a geometric random variable which assumes value k can be generated at cost $O(k+1)$. From this fact, the total cost of a Boltzmann sampler is of the form

$$O\left(\sum_{\nu \in \tau[\omega]} (\deg(\nu) + 1)\right),$$

where the summation ranges over all the nodes ν of tree τ, and $\deg(\nu)$ is the outdegree of node ν. Since, for any tree τ, one has $\sum_\nu 1 = |\tau|$ and $\sum_\nu \deg(\nu) = |\tau| - 1$, the total cost is linear in the size of τ, hence linear in the size of ω. The statement follows. ∎

Given this proposition, one can *compile* automatically specifications of combinatorial classes into Boltzmann samplers. The only piece of auxiliary data required is a table of constants representing the values of the ordinary generating functions associated with the subclasses that intervene in a specification. These are in finite number and computable.

In the examples that follow, we enlarge the expressivity of the specification language by allowing constructions of the form

$$\mathfrak{S}_\Omega(\mathcal{A}) = \left\{ \langle \alpha_1, \ldots, \alpha_r \rangle \ \middle|\ \alpha_j \in \mathcal{A},\ r \in \Omega \right\}, \tag{13}$$

where $\Omega \subset \mathbb{N}$ is either a finite or a cofinite subset of the integers. If Ω is finite, this construction reduces to a disjunction of finitely many cases and the corresponding sampler is obtained by Bernoulli trials. If Ω is cofinite, we may assume without loss of generality that $\Omega = \{n \geq m_0\}$ for some $m_0 \in \mathbb{N}$, in which case, the construction $\mathfrak{S}_{\geq m_0}(\mathcal{A})$ reduces to $\mathcal{A}^{m_0} \times \mathfrak{S}(\mathcal{A})$.

EXAMPLE 1. *Words without long runs.* Consider the collection $\mathcal{R}$ of all binary words over the alphabet $\mathcal{A} = \{a, b\}$ that never have more than m consecutive occurrences of any letter (such consecutive sequences are also called "runs" and intervene at many places in statistics, coding theory, and genetics). Here we regard m as a fixed quantity. It is not *a priori* obvious how to generate a random word in $\mathcal{R}$ of length n: a brutal rejection method based on generating random unconstrained words and filtering out those that satisfy the condition $\mathcal{R}$ will not work in polynomial time since the constrained words have an exponentially small probability. On the other hand, any word decomposes into a sequence of alternations also called its *core*, of the form

$$(aa\cdots a \,|\, bb\cdots b)\,(aa\cdots a \,|\, bb\cdots b)\ \cdots (aa\cdots a \,|\, bb\cdots b), \tag{14}$$

possibly prefixed with a header of b's and postfixed with a trailer of a's. In symbols, the set $\mathcal{W}$ of all words is expressible by a regular expression written in our notation

$$\mathcal{W} = \mathfrak{S}(b) \times \mathfrak{S}\left(a\mathfrak{S}(a)b\mathfrak{S}(b)\right) \times \mathfrak{S}(a).$$

The decomposition was customized to serve for $\mathcal{R}$: simply replace any internal $a\mathfrak{S}(a)$ by $\mathfrak{S}_{1\,..\,m}(a)$ and any $b\mathfrak{S}(b)$ by $\mathfrak{S}_{1\,..\,m}(b)$, where $\mathfrak{S}_{1\,..\,m}$ means a sequence of between 1 and m elements, and adapt accordingly the header and trailer:

$$\mathcal{R} = \mathfrak{S}_{\leq m}(b) \times \mathfrak{S}\left(\mathfrak{S}_{1\,..\,m}(a)\mathfrak{S}_{1\,..\,m}(b)\right) \times \mathfrak{S}_{\leq m}(a).$$

The composition rules given above give rise to a generator for $\mathcal{R}$ that has the following form: two generators that produce sequences of a's or b's according to a truncated geometric law; a generator for the product $\mathcal{C} := (\mathfrak{S}_{1\,..\,m}(a)\mathfrak{S}_{1\,..\,m}(b))$ that is built according to the product rule; a generator for the sequence $\mathcal{D} := \mathfrak{S}(\mathcal{C})$ constructed according to the sequence rule. The generator finally assembled ***automatically*** is:

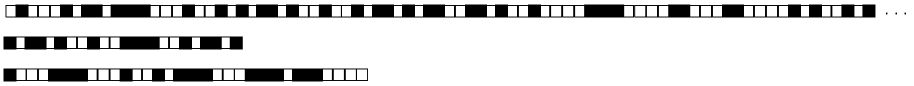
$$\begin{array}{rcl} \Gamma R(x) & = & (X \Longrightarrow b); \Gamma\mathrm{Core}(x); (X' \Longrightarrow a) \\ \Gamma\mathrm{Core}(x) & = & \left(\mathrm{Geom}\left(\frac{x^2(1-x^m)^2}{(1-x)^2}\right) \Longrightarrow ((Y \Longrightarrow a); (Y' \Longrightarrow b))\right) \\ & & X, X' \in \underset{\leq m}{\mathrm{Geom}}(x), \quad Y, Y' \in \underset{1\,..\,m}{\mathrm{Geom}}(x). \end{array}$$

Observe that a table of only a small number of real-valued constants rationally related to x and including

$$c_1 = x, \quad c_2 = C(x) = x^2(1-x^m)^2(1-x)^{-2},$$

needs to be precomputed in order to implement the algorithm. □

Here are three runs of the sampler $\Gamma R(x)$ for $m = 4$ produced with the coherent value $x = 0.5$ (the critical value is $\rho_R \doteq 0.51879$), of respective lengths 124 (truncated), 23, and 35, with the coding $a =$□, $b =$ ■:



With this value of the parameter, the mean size of a random word produced is about 27. The distribution turns out to be of the "flat" type, like for Surjections in Figure 1. We shall see later in Section 7 that one can design optimized samplers for such types of distributions. The technique applies to any language composed of words with excluded patterns, meaning words that are constrained *not* to contain any of a finite set of words as factor. (For such a language, one can specifically construct a finite automaton by way of the Aho–Corasick construction [1], then write the automaton as a linear system of equations relating specifications, and finally compile the set of equations into a recursive Boltzmann sampler.) More generally, the method applies to any regular language: it suffices to convert a description of the language into a deterministic finite automaton and apply the recursive construction of a sampler, or alternatively to obtain an unambiguous regular expression and derive from it a nonrecursive sampler based on the geometric law.

The next set of examples is relative to structures that satisfy nonlinear recursive descriptions of the context-free type.

EXAMPLE 2. *Rooted plane trees.* Take the class $\mathcal{B}$ of binary trees defined by the recursive specification

$$\mathcal{B} = \mathcal{Z} + (\mathcal{Z} \times \mathcal{B} \times \mathcal{B}),$$

where $\mathcal{Z}$ is the class comprising the generic node. The generator ΓZ is deterministic and consists simply of the instruction "output a node" (since $\mathcal{Z}$ is finite and in fact has only one element). The Boltzmann generator ΓB calls ΓZ (and halts) with probability $x/B(x)$ where $B(x)$ is the OGF of binary trees,

$$B(x) = \frac{1 - \sqrt{1-4x^2}}{2x}.$$

With the complementary probability corresponding to the strict binary case, it will make a call to ΓZ and two recursive calls to itself. In shorthand notation, the recursive sampler is

$$\Gamma B(x) = \Big(\text{Bern}\left(\tfrac{x}{B(x)}\right) \longrightarrow \mathcal{Z} \;\Big|\; (\mathcal{Z}; \Gamma B(x); \Gamma B(x))\Big).$$

In other words: *the Boltzmann generator for binary trees as constructed automatically from the composition rules produces a random sample of the branching process with probabilities* $(\frac{x}{B(x)}, \frac{xB(x)^2}{B(x)})$. Note that the generator is defined for $x < 1/2$ (the radius of convergence of $B(x)$), in which case the branching process is subcritical, so that the algorithm halts in finite expected time, as it should. Only two constants are needed for implementation, namely x and the quadratic irrational $\frac{x}{B(x)}$.

Unbalanced 2-3 trees in which only external nodes contribute to size are similarly produced by $\mathcal{U} = \mathcal{Z} + \mathcal{U}^2 + \mathcal{U}^3$. Figure 3 displays such a tree for the value of the parameter x set at the critical value $\rho_U = \frac{5}{27}$. (This critical value can be determined by methods exposed in Section 7.) In this case, the branching probabilities for a nullary, binary, and ternary node are found to be respectively

$$p_0 = \frac{5}{9}, \qquad p_2 = \frac{1}{3}, \qquad p_3 = \frac{1}{9},$$

and these three constants are the only ones required by the algorithm. A typical run of 30 Boltzmann samplings produces trees with total number of nodes equal to

$$(15)\quad 3, 6, 1, 1, 6, 7, 33, 1, 1, 1, 9, 1, 1, 3, 1, 3, 169, 1881, 1, 54, 6, 1, 1, 3, 3746, 1, 1, 1, 1, 1,$$

which empirically gives an indication of the distribution of sizes (it turns out to be of the peaked type, like in Figure 1, bottom). We shall see later in Section 7 that one can actually characterize the profile of this distribution (it decays like $n^{-3/2}$) and put to good use some of its features.

Unary-binary trees (also known as Motzkin trees) are defined by $\mathcal{V} = \mathcal{Z}(\mathbf{1} + \mathcal{V} + \mathcal{V}^2)$. General plane trees, $\mathcal{G}$, where all degrees of nodes are allowed, can be specified by the grammar

$$\mathcal{G} = \mathcal{Z} \times \mathfrak{S}(\mathcal{G}),$$

with OGF $G(z) = (1 - \sqrt{1-4z})/2$. Accordingly, the automatically produced sampler is

$$\Gamma G(x) = (\mathcal{Z}; (\text{Geom}(G(x)) \Longrightarrow \Gamma G(x))),$$

which corresponds to the well-known fact that such trees are equivalent to trees of a branching process where the offspring distribution is geometric. □

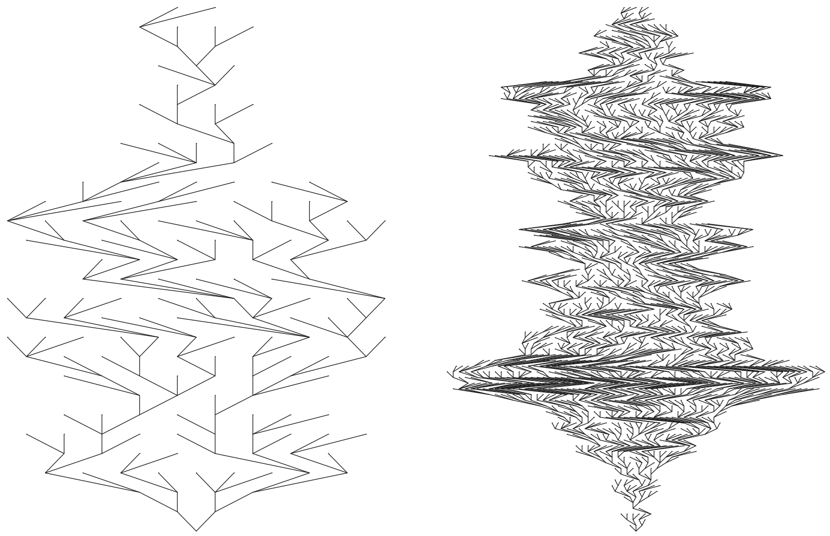


FIGURE 3. Random unbalanced 2–3 trees of 173 and 2522 nodes (in total) produced by a critical Boltzmann sampler.

EXAMPLE 3. *Secondary structures.* This example is inspired by works of Waterman *et al.*, themselves motivated by the problem of enumerating secondary RNA structures [36, 62]. To fix ideas, consider rooted binary trees where edges contain 2 or 3 atoms and leaves ("loops") contain 4 or 5 atoms. A specification is $\mathcal{W} = (\mathcal{Z}^4 + \mathcal{Z}^5) + (\mathcal{Z}^2 + \mathcal{Z}^3)^2 \times (\mathcal{W} \times \mathcal{W})$. A Bernoulli switch will decide whether to halt or not, two independent recursive calls being made in case it is decided to continue, with the algorithm being sugared with suitable Bernoulli draws. Here is the complete code:

$$
\begin{array}{rcl}
\Gamma A(x) & = & \left(\mathrm{Bern}(\frac{x^4}{x^4+x^5}) \longrightarrow Z^4 \mid Z^5\right) \\
\Gamma B(x) & = & \left(\mathrm{Bern}(\frac{x^2}{x^2+x^3}) \longrightarrow Z^2 \mid Z^3\right) \\
& & \text{let } p = (x^4+x^5)/W(x) = \frac{1}{2}(1+\sqrt{1-4x^8(1+x)^3}); \\
\Gamma W(x) & = & \big(\mathrm{Bern}(p) \longrightarrow \Gamma A(x) \mid \Gamma B(x); \Gamma W(x); \Gamma B(x); \Gamma W(x)\big) .
\end{array}
$$

The method is clearly universal for this entire class of problems. □

EXAMPLE 4. *Noncrossing graphs.* Consider graphs which, for size n, have vertices at the nth roots of unity, $v_k = e^{2ik\pi/n}$, and are *connected* and *noncrossing* in the sense that no two edges are allowed to meet in the interior of the unit circle; see Figure 4 for a random instance. The generating function of such graphs has been first determined by Domb and Barret [15] motivated by the investigation of certain perturbative expansions of statistical physics. Their derivation is not based on methods conducive to Boltzmann sampling, though. On the other hand, the planar structure of such configurations entails a neat decomposition, which is described in [24]. At the top level, consider the graph as rooted at vertex v_0. Let v_i and v_j be two consecutive neighbours of v_0; the subgraph induced on the vertex set $\{v_i, v_{i+1}, \ldots, v_j\}$ is either a connected graph of $\mathcal{D}$ or is formed of two disjoint components containing v_i and v_j respectively. Also, if v_ℓ is the first neighbour of v_0 and

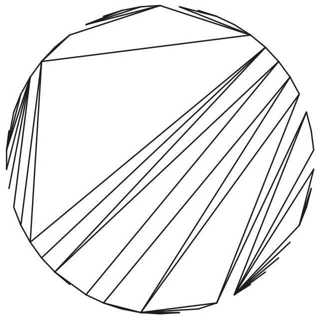


FIGURE 4. A random connected noncrossing graph of size 50.

v_m is the last neighbour, there are two connected components on $\{v_1, \ldots, v_\ell\}$ and on $\{v_m, \ldots, v_{n-1}\}$ respectively. The grammar for connected noncrossing graphs is then a transcription of this simple decomposition, although its detail is complicated as care must be exercised to avoid double counting of vertices. The class of all such connected noncrossing graphs is denoted by $\mathcal{X}$ and the grammar is:

$$\mathcal{X} = \mathcal{Z} + \mathcal{Z} \times \mathcal{E}, \quad \mathcal{E} = \mathcal{X} \times \mathfrak{S}(\mathcal{E} + \mathcal{X} \times (1 + \mathcal{E})) \times \mathcal{X}.$$

One finds that $E(z) = -1 + X(z)/z$ while $X(z)$ is a branch of the algebraic function defined implicitly by

$$X^3 + X^2 - 3zX + 2z^2 = 0,$$

and the critical value (the upper limit of all coherent values) is $\rho_X = \frac{1}{18}\sqrt{3} \doteq 0.09622$. The Boltzmann sampler compiled from the specification is then of the global form

$$\begin{array}{rcl} \Gamma X(x) & = & \left(\mathrm{Bern}\left(\frac{x}{X(x)}\right) \longrightarrow \mathcal{Z} \mid \mathcal{Z}; \Gamma E(x)\right) \\ \Gamma E(x) & = & (\Gamma X(x); (\mathrm{Geom}(E(x) + X(x)(1 + E(x))) \Longrightarrow ((\cdots))) ; \Gamma X(x)) . \end{array}$$

The algorithm needs the parameter x, the cubic quantity $y = X(x)$ and a small number of quantities that are all rationally expressed in terms of x and y. For instance, the coherent choice $x = 0.095$ which is close to the critical value ρ_X, leads to $X(x) \doteq 0.11658$. There is then a probability of about $\frac{1}{7000}$ to attain a graph of size exactly 50; one such graph drawn uniformly at random is represented in Figure 4. $\square$

In the last three cases (trees, secondary structures, and noncrossing graphs), the profile of the Boltzmann distribution resembles that of general trees in Figure 1. Optimized algorithms adapted to such tree-like profiles are discussed in Sections 6 and 7, where it is shown that random generation can be achieved in linear time provided a fixed nonzero tolerance on size is allowed. The method applies to any class that can be described unambiguously by a context-free grammar.

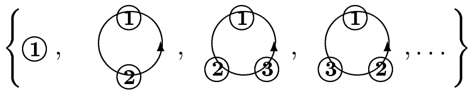
4. EXPONENTIAL BOLTZMANN GENERATORS

We consider here *labelled structures* in the precise technical sense of combinatorial theory [4, 28, 30, 34, 60, 61, 69]. A labelled object of size n is then composed of n distinguishable atoms, each bearing a distinctive label that is an integer in the interval $[1, n]$. For instance, the class $\mathcal{K}$ of labelled circular graphs, where cycles

Construction		*Generator*
Singleton	$\mathcal{C} = \{\omega\}$	$\Gamma C(x) = \omega$
Union	$\mathcal{C} = \mathcal{A} + \mathcal{B}$	$\Gamma C(x) = \left(\text{Bern}\left(\frac{\widehat{A}(x)}{\widehat{A}(x)+\widehat{B}(x)}\right) \longrightarrow \Gamma A(x) \mid \Gamma B(x)\right)$
Product	$\mathcal{C} = \mathcal{A} \star \mathcal{B}$	$\Gamma C(x) = \big(\Gamma A(x); \Gamma B(x)\big)$
Sequence	$\mathcal{C} = \mathfrak{S}(\mathcal{A})$	$\Gamma C(x) = \big(\text{Geom}(\widehat{A}(x)) \Longrightarrow \Gamma A(x)\big)$
Set	$\mathcal{C} = \mathfrak{P}(\mathcal{A})$	$\Gamma C(x) = \big(\text{Pois}(\widehat{A}(x)) \Longrightarrow \Gamma A(x)\big)$
Cycle	$\mathcal{C} = \mathfrak{C}(\mathcal{A})$	$\Gamma C(x) = \big(\text{Loga}(\widehat{A}(x)) \Longrightarrow \Gamma A(x)\big)$

FIGURE 5. The inductive rules for exponential Boltzmann samplers.

are oriented in some conventional manner (say, positively) is

$\mathcal{K} =$ 

Clearly, there are $K_n = (n-1)!$ labelled objects of size $n \geq 1$, and the corresponding exponential generating function $\widehat{K}(z)$ has been determined in (3). In what follows, we focus on generating the "shape" of labelled objects—for instance, the shape of an n-cyclic graph would be a cycle with n anonymous dots placed on it. The reason for doing so is that labels can then always be obtained by superimposing a random permutation[4] on the unlabelled nodes. Note however, that the unlabelled (ordinary) and labelled (exponential) Boltzmann models assign rather different probabilities to objects: in the unlabelled case, there would be only $k_n \equiv 1$ object of size n, with OGF $k(x) = x/(1-x)$ so that the distribution of component sizes is geometric, while in the labelled case, the logarithmic series distribution (4) occurs.

Labelled combinatorial classes can be subjected to the *labelled product* defined as follows: if $\mathcal{A}$ and $\mathcal{B}$ are labelled classes, the product $\mathcal{C} = \mathcal{A} \star \mathcal{B}$ is obtained by forming all ordered pairs $\langle \alpha, \beta \rangle$ with $\alpha \in \mathcal{A}$ and $\beta \in \mathcal{B}$ and relabelling them in all possible order-consistent ways. Straight from the definition, one has a binomial convolution $C_n = \sum_{k=0}^{n} \binom{n}{k} A_k B_{n-k}$, where the binomial takes care of relabellings. In terms of exponential generating functions, this becomes

$$\widehat{C}(z) = \widehat{A}(z) \cdot \widehat{B}(z).$$

Like in the ordinary case, we proceed by assembling Boltzmann generators for structured objects from simpler ones.

Disjoint union. The unlabelled construction carries over verbatim to this case to the effect that, for labelled classes $\mathcal{A}, \mathcal{B}, \mathcal{C}$ satisfying $\mathcal{C} = \mathcal{A} + \mathcal{B}$, EGFs are related by $\widehat{C}(z) = \widehat{A}(z) + \widehat{B}(z)$ and the exponential Boltzmann sampler for C is

$$\Gamma C(x) = \left(\text{Bern}\left(\frac{\widehat{A}(x)}{\widehat{A}(x)+\widehat{B}(x)}\right) \longrightarrow \Gamma A(x) \mid \Gamma B(x)\right).$$

Labelled product. The cartesian product construction adapts to this case with minor modifications: to produce an element from $\mathcal{C} = \mathcal{A} \star \mathcal{B}$, simply produce a pair

[4] Drawing a random permutation of $[1, n]$ only necessitates $O(n)$ real operations [39, p. 145].

by the cartesian product rule using values $\widehat{A}(x), \widehat{B}(x)$:

$$\Gamma C(x) = (\Gamma A(x); \Gamma B(x)) .$$

Complete by a randomly chosen relabelling if actual values of the labels are needed.

Sequences. In the labelled universe, $\mathcal{C}$ is the sequence class of $\mathcal{A}$, written $\mathcal{C} = \mathfrak{S}(\mathcal{A})$ iff it is composed of all the sequences of elements from A up to order-consistent relabellings. Then, the EGF relation

$$\widehat{C}(x) = \sum_{k \geq 0} \widehat{A}(x)^k = \frac{1}{1 - \widehat{A}(x)}$$

holds, and either of the two constructions of the generator ΓC from ΓA given in Section 3 is applicable. In particular, the nonrecursive generator is

$$\Gamma C(x) = \big(\,\mathrm{Geom}(\widehat{A}(x)) \Longrightarrow \Gamma A(x)\big),$$

where the stenographic convention of (12) is employed.

Sets. This is a new construction that we did not consider in the unlabelled case. The class $\mathcal{C}$ is the set-class of $\mathcal{A}$, written $\mathcal{C} = \mathfrak{P}(\mathcal{A})$ ($\mathfrak{P}$ is reminiscent of "powerset") if $\mathcal{C}$ is the quotient of sequences, $\mathcal{C} = \mathfrak{S}(\mathcal{A})/\equiv$, by the relation $\equiv$ that declares two sequences as equivalent if one derives from the other by an arbitrary permutation of the components. It is then easily seen that the EGFs are related by

$$\widehat{C}(x) = \sum_{k \geq 0} \frac{1}{k!} \widehat{A}(x)^k = e^{\widehat{A}(x)},$$

where the factor $1/k!$ "kills" the order present in k–sequences.

The Poisson law of rate λ is classically defined by

$$\mathbb{P}(X = k) = e^{-\lambda} \frac{\lambda^k}{k!}.$$

On the other hand, under the exponential Boltzmann, the probability for a set in $\mathcal{C}$ to have k components in $\mathcal{A}$ is

$$\frac{1}{\widehat{C}(x)} \frac{1}{k!} \widehat{A}(x)^k = e^{-\widehat{A}(x)} \frac{\widehat{A}(x)^k}{k!},$$

that is, a Poisson law of rate $\widehat{A}(x)$. This gives rise to a simple algorithm for generating sets (analogous to the geometric algorithm for sequences):

$$\Gamma C(x) = \big(\,\mathrm{Pois}(\widehat{A}(x)) \Longrightarrow \Gamma A(x)\big).$$

Cycles. This construction, written $\mathcal{C} = \mathfrak{C}(\mathcal{A})$, is defined like sets but with two sequences being identified if one is a cyclic shift of the other. The EGFs satisfy

$$\widehat{C}(x) = \sum_{k \geq 1} \frac{1}{k} \widehat{A}(x)^k = \log \frac{1}{1 - \widehat{A}(x)},$$

where the factor $1/k$ "converts" k–sequences into k–cycles. The log-law of rate $\lambda < 1$, an "integral" of the geometric law also known as the logarithmic series distribution, is the law of a variable X such that

$$\mathbb{P}(X = k) = \frac{1}{\log(1 - \lambda)^{-1}} \frac{\lambda^k}{k}.$$

(This is the same as in Equation (4); the distribution occurs in statistical ecology and economy and forms the subject of Chapter 7 of [38].) Then cycles under the exponential Boltzmann model can be drawn like in the case of sets upon replacing the Poisson law by the log-law:

$$\Gamma C(x) = \big(\operatorname{Loga}(\widehat{A}(x)) \Longrightarrow \Gamma A(x)\big).$$

These constructions are summarized in Figure 5.

For reasons identical to the ones that justify Theorem 1, one has:

Theorem 2. *Define as* specifiable *a labelled class that can be* finitely *specified (in a possibly recursive way) from finite sets by means of disjoint unions, cartesian products, as well as the sequence, set and cycle constructions. Let $\mathcal{C}$ be a labelled specifiable class and x be a coherent parameter in $(0, \rho_C)$. Assume as given an oracle that provides the finite collection of* exact *values at a coherent value x of the generating functions intervening in a specification of a class $\mathcal{C}$. Then, the Boltzmann generator $\Gamma C(x)$ assembled from the definition of $\mathcal{C}$ by means of the six rules of Figure 5 has a complexity measured in the number of $(+, -, \times, \div)$ real-arithmetic operations that is* linear *in the size of its output object.*

(We also allow constructions $\mathfrak{S}_\Omega, \mathfrak{P}_\Omega, \mathfrak{C}_\Omega$ as in (13); in this case, the random variable of geometric, Poisson, or logarithmic type should be conditioned to assume its values in the set Ω.)

Like in the unlabelled case, Boltzmann samplers can be compiled automatically from combinatorial specifications. There is here added expressivity in the language of specifications, thanks to the inclusion of the Set and Cycle constructions. In the examples that follow, we omit the hat-marker "$\widehat{f}$", whenever the exponential/labelled character of the model is clear from the context.

EXAMPLE 5. *Set partitions.* A set partition of size n is a partition of the integer interval $[1, n]$ into a certain number of nonempty classes, also called blocks, the blocks being by definition unordered between themselves. Let $\mathfrak{P}_{\geq 1}$ represent the powerset construction where the number of components is constrained to be ≥ 1. (This modified construction is easily subjected to random generation by using a truncated Poisson variable K, where K is conditioned to be ≥ 1.) The labelled class of all set partitions is then definable as $\mathcal{S} = \mathfrak{P}(\mathfrak{P}_{\geq 1}(\mathcal{Z}))$, where $\mathcal{Z}$ consists of a single labelled atom, $\mathcal{Z} = \{1\}$. Observe that the EGF of $\mathcal{S}$ is the well-known generating function of the Bell numbers, $S(z) = e^{e^z - 1}$. By the composition rules, one gets a random generator as follows: *Choose the number K of blocks as Poisson($e^x - 1$). Draw K independent copies $X_1, X_2, \ldots, X_K$ from the Poisson law of rate x, but each conditioned to be at least 1.* In symbols:

$$\Gamma S(x) = \left(\operatorname{Pois}(e^x - 1) \Longrightarrow \left(\operatorname*{Pois}_{\geq 1}(x) \Longrightarrow \mathcal{Z}\right)\right).$$

What this generates is in reality the "shape" of a set partition (the number of blocks (K) and the block sizes (X_j)), with the "correct" distribution. To complete the task, it suffices to transport this structure on a random permutation of the integers between 1 and N, where $N = X_1 + \cdots + X_K$.

The process distinctly differs from the classical algorithm of Nijenhuis and Wilf [51] that requires tables of large integers. It is related to a continuous model devised by Vershik [67] that can be interpreted as generating random set partitions based on

$$S(x) = e^{x/1!} \cdot e^{x^2/2!} \cdot e^{x^3/3!} \cdots,$$

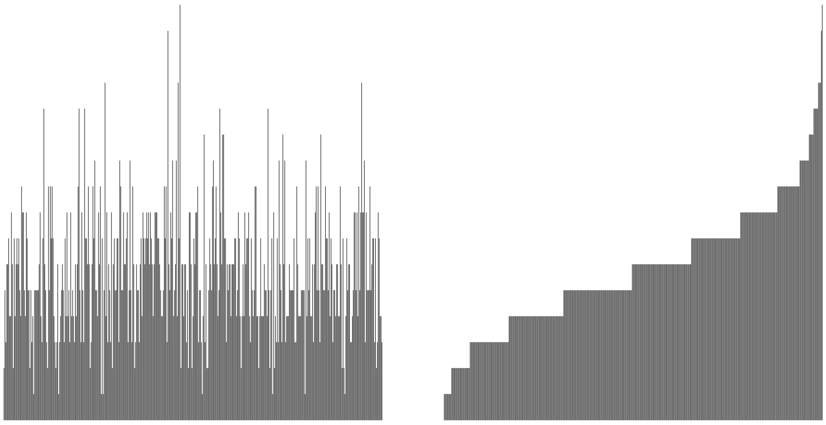


FIGURE 6. A random partition obtained by the Boltzmann parameter of parameter $x = 6$, here of size $n = 2356$ and comprised of 409 blocks: (left) the successive block sizes generated; (right) the block sizes in sorted order.

i.e., by ordered block lengths, as a potentially infinite sequence of Poisson variables of parameters $x/1!$, $x^2/2!$, and so on. $\square$

Figure 6 represents a random set partition produced by the Boltzmann model of parameter $x = 6$. This particular object has size $n = 2356$, the expected size being $\mathbb{E}_x(N) = 2420$ for this value of the parameter. The closeness between the observed size and its mean value agrees with the concentration that is perceptible on Figure 1. In addition, the Boltzmann model immediately provides a simple heuristic model of partitions of large size. Objects of size "near" n, are produced by the value x_n defined by $x_n e^{x_n} = n$, that is, $x_n \approx \log n - \log\log n$. Then, the number of blocks is expected to be about $e^{x_n} \approx n/(\log n)$. This number being large, and individual blocks being generated by independent Poisson variables of parameter x_n, we expect, for large n, the sorted profile of blocks to converge to the histogram of the Poisson distribution of rate x_n (Figure 6, right). As shown by Vershik [67], this heuristic model is indeed a valid asymptotic model of partitions of large sizes.

EXAMPLE 6. *Random surjections (or ordered set partitions).* These may be defined as functions from $[1, n]$ to $[1, n]$ such that the image of f is an initial segment of $[1, n]$ (i.e., there are no "gaps"). One has for the class $\mathcal{Q}$ of surjections $\mathcal{Q} = \mathfrak{S}(\mathfrak{P}_{\geq 1}(\mathcal{Z}))$. Thus a random generator for $\mathcal{Q}$ is:

$$\Gamma Q(x) = \left(\text{Geom}(e^x - 1) \Longrightarrow \left(\underset{\geq 1}{\text{Pois}}(x) \Longrightarrow \mathcal{Z}\right)\right).$$

In words: first choose a number of components given by a geometric law and then launch a number of Poisson generators conditioned to be at most 1. $\square$

Set partitions find themselves attached to a compound (Poisson∘Poisson) process, whereas surjections are generated by a compound (Geometric∘Poisson) process (with suitable dependencies on parameters). This reflects the basic combinatorial opposition between freedom and order (for blocks). Here are two more examples.

EXAMPLE 7. *Cycles in permutations.* This corresponds to $\mathcal{P} = \mathfrak{P}(\mathfrak{C}_{\geq 1}(\mathcal{Z}))$ and is obtained by a (Poisson∘Log) process:

$$\Gamma P(x) = \left(\text{Pois}(\log(1-x)^{-1}) \Longrightarrow (\text{Loga}(x) \Longrightarrow \mathcal{Z})\right).$$

This example is loosely related to the Shepp–Lloyd model [57] that generates permutations by ordered cycle lengths, as a potentially infinite sequence of Poisson variables of parameters $x/1$, $x^2/2$, and so on. The interest of this construction is to give rise to a number of useful particularizations. For instance derangements (permutations such that $\sigma(x) \neq x$) are produced by $\mathcal{P} = \mathfrak{P}(\mathfrak{C}_{\geq 2}(\mathcal{Z}))$; involutions (permutations such that $\sigma \circ \sigma(x) = x$) are given by $\mathcal{P} = \mathfrak{P}(\mathfrak{C}_{1\,..\,2}(\mathcal{Z}))$. □

EXAMPLE 8. *Assemblies of filaments.* Imagine assemblies of linear filaments floating freely in a liquid. We may model these as sets of sequences, $\mathcal{F} = \mathfrak{P}(\mathfrak{S}_{\geq 1}(\mathcal{Z}))$. The EGF is $\exp\left(\frac{z}{1-z}\right)$. The random generation algorithm is a compound of the form (Poisson∘Geometric), with appropriate parameters:

$$\Gamma F(x) = \left(\text{Pois}\left(\tfrac{x}{1-x}\right) \Longrightarrow \left(\underset{\geq 1}{\text{Geom}}(x) \Longrightarrow \mathcal{Z}\right)\right).$$

The corresponding counting sequence, $1, 1, 3, 13, 73, 501, \ldots$, appears as A000262 in Sloane's encyclopedia [58]. This example is closely related to linear forests and posets as described in Burris' book (see [6], pp. 23–24 and Ch. 4). □

At this stage, it may be of interest to note that many classical probabilistic distributions of probability theory can be retrieved as (size distributions of) Boltzmann models associated to simple combinatorial games. Consider an unbounded supply of distinguishable (i.e., labelled) balls. View an urn as an unordered finite collection of balls ($\mathfrak{P}(\mathcal{Z})$) and a stack as an ordered collection of balls ($\mathfrak{S}(\mathcal{Z})$). The geometric and Poisson distributions arise as the size distributions of the stack and the urn. If, by an exclusion principle, an urn is only allowed to contain 0 or 1 ball ($\mathbf{1}+\mathcal{Z}$), then the family of all basic Bernoulli distributions results. If m urns or stacks are considered, then the distributions are Poisson or negative binomial, respectively, and, with exclusion, one gets in this way the binomial distributions corresponding to m trials. If balls and urns are taken to be indistinguishable, one obtains automatically Vershik's model of integer partitions [67], which is an infinite product of geometric distributions of exponentially decaying rates. (The recent work by Milenković and Compton [50] discusses exact and asymptotic transforms associated to several such distributions.) For similar reasons, the two classical models of random graphs due to Erdős and Rényi are related to one another by "Boltzmannization". A large number of examples along similar lines could clearly be listed.

5. THE REALIZATION OF BOLTZMANN SAMPLERS

In this section, we make explicit the way Boltzmann sampling can be implemented and sketch a discussion of the main complexity issues involved. Broadly speaking, samplers can be realized under two types of computational models corresponding to computations carried out over the set $\mathbb{R}$ of real numbers or the set $\mathbb{S} = \{0,1\}^{\mathbb{N}}$ of infinite-length binary strings. (In the latter case, only finite prefixes are ever used.) These are the *real-arithmetic model*, $\mathbb{R}$, which is the one considered here and the *bit string model* (or Boolean model), $\mathbb{S}$, whose algorithms will be described in a future publication. The "ideal" real-domain model $\mathbb{R}$ comprises the elementary operations $+, -, \times, \div$ each taken to have *unit cost*.

By definition, a Boltzmann sampler requires as input the value of the control parameter x that defines the Boltzmann model of use. As seen in previous sections, it also needs the finite collection of values at x of the generating functions that intervene in a specification, in order to drive Bernoulli, geometric, Poisson, and logarithmic generators. We assume these values to be provided by what we call the (generating function) "*oracle*":

$$\boxed{\begin{array}{c}\text{ORACLE}\\ x \rightsquigarrow C(x), \ldots\end{array}} \longrightarrow \boxed{\boxed{\begin{array}{c}\text{SAMPLER}\\ \Gamma C(x)\end{array}}}$$

Such constants need only be precomputed *once*; they can be provided by a multi-precision package or a computer algebra system used as coroutine. We take here these constants as given, noting that the corresponding power series expansions at 0 are computable in low polynomial complexity (this is, e.g., encapsulated in the Maple package Combstruct; see [27, 29] for the underlying principles) so that values of the generating functions of constructible classes strictly inside their disc of convergence are computable real numbers of low polynomial time complexity.

There remains to specify fully generators for the probabilistic laws $\mathrm{Geom}(\lambda)$, $\mathrm{Pois}(\lambda)$, $\mathrm{Loga}(\lambda)$, as well as the Bernoulli generator $\mathrm{Bern}(p)$, where the latter outputs 1 with probability p and 0 otherwise. What is assumed here is a random generator 'uniform()' that produces at unit cost a random variable uniformly distributed over the real interval $(0, 1)$.

Bernoulli generator. The Bernoulli generator is simply

$$\mathrm{Bern}(p) := \textsf{if } \mathrm{uniform}() \le p \textsf{ then } \mathrm{return}(1) \textsf{ else } \mathrm{return}(0) \textsf{ fi}.$$

This generator serves in particular to draw from unions of classes.

Geometric, Poisson, and Logarithmic generators. For the remaining laws, we let p_k be the probability that a random variable with the desired distribution has value k, namely,

$$\mathrm{Geom}(\lambda) : p_k = (1-\lambda)\lambda^k; \ \mathrm{Pois}(\lambda) : p_k = e^{-\lambda}\frac{\lambda^k}{k!}; \ \mathrm{Loga}(\lambda) : p_k = \frac{1}{\log(1-\lambda)^{-1}}\frac{\lambda^k}{k}.$$

The general scheme that goes well with real-arithmetic models is the *sequential algorithm*:

$U := \mathrm{uniform}();\ S := 0;\ k := 0;$
$\textsf{while } U < S \textsf{ do } S := S + p_k;\ k := k + 1; \textsf{ od};$
$\mathrm{return}(k).$

This scheme is nothing but a straightforward implementation based on *inversion* of distribution functions (see [14, Sec. 2.1] or [39, Sec. 3.4.1]). For the three distributions under consideration, the probabilities p_k can themselves be computed recurrently on the fly as follows:

	$\mathrm{Geom}(\lambda)$	$\mathrm{Pois}(\lambda)$	$\mathrm{Loga}(\lambda)$
(16)	$p_0 = (1-\lambda)$	$p_0 = e^{-\lambda}$	$p_1 = 1/\left(\log(1-\lambda)^{-1}\right)$
	$p_{k+1} = \lambda p_k$	$p_{k+1} = \lambda p_k \frac{1}{k+1}$	$p_{k+1} = \lambda p_k \frac{k}{k+1}$.

(Such principles also apply to constructions modified by a constraint on the number of components; e.g., to generate a $\mathrm{Pois}_{\ge 1}(\lambda)$ random variable, initialize the generator with $p_1 = (e^{\lambda} - 1)^{-1}$ and $k = 1$.)

Observe that the transcendental values in (16) (like $e^{-\lambda}$) are in the present context already provided by the oracle. For instance, if one has to generate sets corresponding to $\mathcal{C} = \mathfrak{P}(A)$, then the generator for sets, $\text{Pois}(A(x)) \Longrightarrow \Gamma A(x)$, requires the knowledge of $e^{-A(x)}$ which is none other than $1/C(x)$. Under the model that has unit cost for the four elementary real-arithmetic operations, the sequential generators thus have a useful property:

Lemma 1. *For either of the geometric, Poisson, or logarithmic generators, a random variable with outcome k is drawn with a number of real-arithmetic operations which is $O(k+1)$.*

This lemma completes te justification of Theorems 1 and 2.

In practice, one may realize approximately a Boltzmann sampler by truncating real numbers to some fixed precision, say using floating point numbers represented on 64 bits or 128 bits. The resulting samplers operate in time that is linear in the size of the object produced, though they may fail (by lack of digits in values of generating functions, i.e., by insufficient accuracy in parameter values) in a small number of cases, and accordingly must deviate (slightly) from uniformity. Pragmatically, such samplers are likely to suffice for many simulations.

A sensitivity analysis of truncated Boltzmann samplers would be feasible, though rather heavy to carry out. One could even correct perfectly the lack of uniformity by appealing to an adaptive precision strategy based on guaranteed multiprecision floating point arithmetic—-e.g., double the accuracy of computations when more digits are needed. In case of floating-point implementations of the recursive method, such ideas are discussed in Zimmermann's survey [71], and the reader may get a feeling of the type of analysis involved by referring to the works of Denise, Dutour, and Zimmermann [12, 13]. In a companion paper, we shall explore another route and describe purely discrete Boltzmann samplers which are solely based on binary coin flips in the style of Knuth and Yao's work [40] and have the additional feature of "automatically" detecting when accuracy is insufficient.

6. Exact-size and approximate-size sampling

Our primary objective in this article is the fast random generation of objects of some large size. In this section and the next one, we consider two types of constraints on size.

- — *Exact-size* random sampling, where objects of $\mathcal{C}$ should be drawn *uniformly* at random from the subclass $\mathcal{C}_n$ of objects of size exactly n.
- — *Approximate-size* random sampling, where objects should be drawn with a size in an interval of the form $I(n,\varepsilon) = [n(1-\varepsilon), n(1+\varepsilon)]$, for some quantity $\varepsilon \geq 0$ called the (relative) *tolerance*. In applications, one is likely to consider cases where ε is a small fixed number, like 0.05, corresponding to an uncertainty on sizes of $\pm 5\%$. Though size may fluctuate (within limits), sampling is still *unbiased* in the sense that two objects with the same size are drawn with equal likelihood.

The conditions of exact and approximate-size sampling are automatically satisfied if one filters the output a Boltzmann generator by retaining only the elements that obey the desired size constraint. (As a matter of fact, we have liberally made use of this feature in previous examples, e.g., when selecting the trees of Figure 3 to be large enough.) Such a filtering is simply achieved by a *rejection* technique.

The main question then becomes: "When and how can the rejection strategy be reasonably efficient?".

The major conclusion of this section is that in many cases, including all the examples seen so far, approximate-size sampling is achievable in linear time under the (exact) real-arithmetic model. In addition, the constants appear to be not too large if a "reasonable" tolerance on size is accepted. Precisely, we develop analyses and optimizations corresponding to the three common types of distributions exemplified in Figure 1.

— For size distributions that are "bumpy", the straight rejection strategy succeeds with high probability in one trial, hence the linear-time complexity of approximate-size Boltzmann sampling results (Section 6.1).
— For size distributions that are "flat", the straight rejection strategy succeeds in $O(1)$ trials on average, a fact that again ensures linear-time complexity when a nonzero tolerance on size is allowed (Section 6.2).
— For size distributions that are "peaked" (at the origin), the technique of *pointing* may be used to transform automatically specifications into equivalent ones of the flat type (Section 6.3).

6.1. **Size-control and rejection samplers.** The basic *rejection sampler* denoted by $\mu C(x; n, \varepsilon)$ uses a Boltzmann generator $\Gamma C(x)$ for the class $\mathcal{C}$ and is described as follows, for any x with $0 < x < \rho_C$, n a target size and $\varepsilon \geq 0$ a relative tolerance:

function $\mu C(x; n, \varepsilon)$;
{Returns an object of $\mathcal{C}$ in $I(n, \varepsilon) := [n(1-\varepsilon), n(1+\varepsilon)]$}
repeat $\gamma := \Gamma C(x)$ until $|\gamma| \in I(n, \varepsilon)$;
return(γ); end.

The rejection sampler μC depends on a parameter x that one may choose arbitrarily amongst all coherent values. It simply tries repeatedly until an object of satisfactory size is produced. The case $\varepsilon = 0$ then gives exact-size sampling.

The outcome of a basic Boltzmann sampler has a random size N whose distribution is described by Proposition 1. One has

$$\mathbb{E}_x(N) = \nu_1(x), \qquad \mathbb{E}_x(N^2) = \nu_2(x), \qquad \mathbb{E}_x(N^2) - \mathbb{E}_x(N)^2 = \sigma(x)^2,$$

where σ represents standard deviation, with

$$\nu_1(x) := x\frac{C'(x)}{C(x)}, \qquad \nu_2(x) := x^2\frac{C''(x)}{C(x)} + x\frac{C'(x)}{C(x)}, \qquad \sigma(x) = \sqrt{\nu_2(x) - \nu_1(x)}.$$

If x stays bounded away from the critical value ρ_C, then $\nu_1(x)$ remains bounded, so that the object drawn is likely to have a small size (on average and in probability). Thus, values of x approaching the critical value $\rho \equiv \rho_C$ have to be considered. Introduce the *mean value condition* as

$$\text{Mean Value Condition :} \quad \lim_{x \to \rho^-} \nu_1(x) = +\infty. \tag{17}$$

(This condition is satisfied in particular when $C(\rho^-) = +\infty$.) Then a "natural tuning" for the rejection sampler consists in adopting as control parameter x the value x_n that satisfies

$$x_n \quad \text{is the root in } (0, \rho) \text{ of} \quad n = x\frac{C'(x)}{C(x)}, \tag{18}$$

which is uniquely determined. One then has:

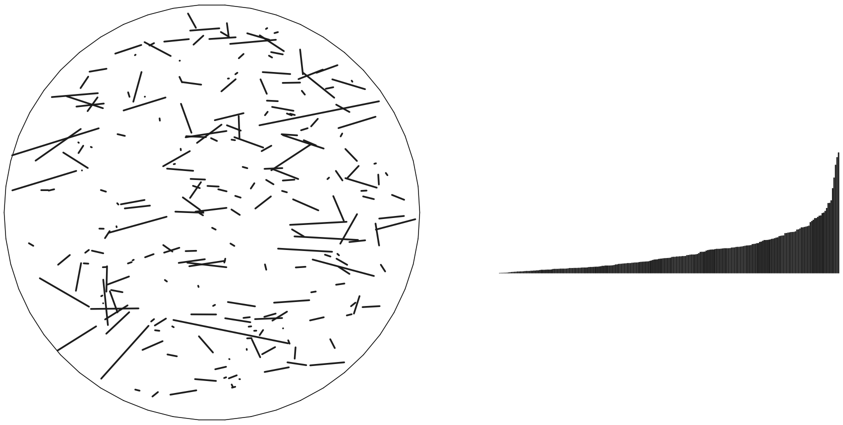


FIGURE 7. A random assembly of filaments of size $n = 46299$ produced by the exponential Boltzmann sampler tuned to $x_{50000} \doteq 0.9952$ (left) and its filaments presented in increasing order of lengths (right).

Theorem 3. *Let $\mathcal{C}$ be a combinatorial class and ε a fixed (relative) tolerance on size. Assume the Mean Value Condition* (17) *and the following variance condition*

$$\text{Variance Condition}: \quad \lim_{x \to \rho^-} \frac{\sigma(x)}{\nu_1(x)} = 0. \tag{19}$$

Then, the rejection sampler $\mu C(x_n; n, \varepsilon)$ equipped with the value $x = x_n$ implicitly determined by (18) *succeeds in one trial with probability tending to 1 as $n \to \infty$. In particular, if $\mathcal{C}$ is specifiable, then the overall cost of approximate-size sampling is $O(n)$ on average.*

Proof. This is a direct consequence of Chebyshev's inequalities. ∎

The mean and variance conditions *are* satisfied by the class $\mathcal{S}$ of set partitions (Example 5, observe concentration on Figure 1, top) and the class $\mathcal{F}$ of assemblies of filaments (Example 8 and Figure 7). In effect, for set partitions, $\mathcal{S}$, the exponential generating function is entire, which corresponds to $\rho = +\infty$. One finds

$$\nu_1(x) = xe^x, \qquad \sigma(x)^2 = x(x+1)e^x,$$

while x_n determined implicitly by the equation $x_n e^{x_n} = n$ satisfies $x_n \sim \log n - \log\log n$. These quantities are most easily interpreted when expressed in terms of n itself:

$$\nu_1(x_n) = n, \qquad \sigma(x_n) \sim \sqrt{n \log n}.$$

For assemblies of filaments, $\mathcal{F}$, one finds $\rho = 1$ and $\nu_1(x) = \frac{x}{(1-x)^2}$, so that x_n has value

$$x_n = 1 + \frac{1}{2n} - \frac{\sqrt{1+4n}}{2n} \sim 1 - \frac{1}{\sqrt{n}}.$$

and $\sigma(x_n) \sim \sqrt{2n}$. Here is, for various values of n, a table of the sizes of objects drawn in batches of 10 runs and the interval in which sizes are found to lie:

n	x_n	N (batch of 10 runs)	$N_{\min}$–$N_{\max}$
50	0.85857	$61, 80, 62, 13, 32, 65, 21, 34, 67, 16$	$13-80$
500	0.95527	$647, 426, 323, 752, 599, 457, 505, 318, 358, 424$	$318-752$
$5,000$	0.98585	$4575, 4311, 4419, 4257, 4035, 4067, 4187, 4984, 4543, 5035$	$4035-5035$

The fact that concentration of distribution improves with larger values of n is perceptible on such data. This feature in turn implies sampling in linear time, as soon as a positive tolerance on size is granted.

Exact-size sampling. The previous discussion suggests investigating conditions under which exact-size generation is still reasonably efficient. The smooth aspect of the "bumpy" curves associated with set partitions suggests the possibility that, in such cases, there exist a local limit distribution for sizes, as $x \to \rho$, implying an expected cost of $O(n\sigma(x_n))$ for exact-size sampling. It turns out that a sufficient set of *complex-analytic* conditions can be stated by borrowing results from the theory of *admissibility*, an area originally developed for the purpose of estimating asymptotically Taylor coefficients of entire functions. This theory was started in an important paper of Hayman [35] and is lucidly exposed in Odlyzko's survey [52, Sec. 12]. A function is said to be H-admissible if, in addition to the mean value condition (17) and the variance condition (19), it satisfies the following two properties:

- There exists a function $\delta(x)$ defined for $x < \rho$ with $0 < \delta(x) < \pi$ such that, for $|\theta| < \delta(x)$ as $x \to \rho^-$,
$$f(xe^{i\theta}) \sim f(x)e^{ia\theta - \frac{1}{2}b\theta^2}, \qquad a = \nu_1(x), \quad b = \sigma^2(x).$$
- Uniformly as $x \to \rho^-$, for $\delta(x) \le |\theta| \le \pi$,
$$f(xe^{i\theta}) = o\left(\frac{f(x)}{\sigma(x)}\right).$$

These conditions are the minimal ones that guarantee the applicability of the saddle-point method to Cauchy coefficient integrals. They imply in particular knowledge of the asymptotic form of the coefficients of f, namely,
$$f_n \equiv [z^n]f(z) \sim \frac{f(x_n)}{\sqrt{2\pi}x_n^n\sigma(x_n)}, \qquad n \to \infty.$$
We state:

Theorem 4. *Consider a class $\mathcal{C}$ whose generating function $f(z)$ satisfies the complex-analytic conditions of H-admissibility. Then exact size rejection sampling base on $\mu C(x_n; n, 0)$ succeeds in a mean number of trials that is asymptotic to*
$$\sqrt{2\pi}\sigma(x_n).$$
In particular, if $\mathcal{C}$ is specifiable, then the overall cost of exact-size sampling is $O(n\sigma(x_n))$ on average.

Proof. This is a direct adaptation of one of Hayman's estimates, see Theorem I of [35] (specialized in the notations of [35] as $r \to x_n$, $n \mapsto m$),
$$\frac{f_m x_n^m}{f(x_n)} \sim \frac{1}{\sqrt{2\pi}\sigma(x_n)} \exp\left(-\frac{(m-n)^2}{2\sigma(x_n)^2} + o(1)\right),$$

uniformly for all m as $x_n \to \rho$. This last equation means generally that the distribution of size values m is asymptotically normal as $x_n \to \rho^-$, that is, as $n \to \infty$. The specialization $m = n$ gives the statement. ∎

Hayman admissibility is easily checked to be satisfied by the EGFs of set partitions and assemblies of filaments. There results that exact size sampling has the following costs:

$$\text{Set partitions}: \quad O(n^{3/2}\sqrt{\log n}); \qquad \text{Assemblies}: \quad O(n^{3/2}).$$

Another result of Hayman states that, under H-admissibility, standard deviation is smaller than the mean, $\sigma(x_n) = o(n)$ (see Corollary I of [35]), so that exact-size generation by Boltzmann rejection is necessarily *subquadratic* ($o(n^2)$).

The usefulness of Hayman's conditions devolves from a rich set of closure properties: under mild restrictions, admissible functions are closed under sum $(f+g)$, product (fg), and exponentiation (e^f). An informally stated consequence is then: *For classes whose generating function is "dominated" by an exponential, i.e., the "principal" construction is of the set type, approximate-size generation is of linear time complexity and exact-size generation is of subquadratic complexity.* Here are a few more examples.

- Statistical classification theory superimposes a tree structure on objects based on a similarity measure (e.g., the number of common phenotypes or genes). In this context, the value of a proposed classification tree may be assessed by comparing it to a random classification tree (structural properties should be substantially different in order for the classification to be likely to make sense). Such comparisons in turn benefit from random generation algorithms, a point originally made by Van Cutsem and collaborators [63, 64]. For instance, *hierarchies* are labelled objects determined by
$$\mathcal{H} = \mathcal{Z} + \mathfrak{P}_{\geq 2}(\mathcal{H}),$$
and they correspond to Schröder's systems of combinatorial theory [9, p. 223–224]. Hierarchies with a bounded depth of nesting are of interest in this context, and their EGFs
$$e^z - 1, \quad z + e^{e^z-1} - e^z, \quad e^{z+e^{e^z-1}-e^z} - 1 - e^{e^z-1} + e^z, \quad \ldots,$$
are all admissible, hence amenable to the conclusions of Theorem 4.
- Similar comments apply to labelled trees (Cayley trees, $\mathcal{T} = \mathcal{Z} \star \mathfrak{P}(\mathcal{T})$) of bounded height, with the sequence of EGFs starting as
$$z, \quad ze^z, \quad ze^{ze^z}, \quad ze^{ze^{ze^z}}, \quad ,\ldots,$$
and to "superpartitions" obtained by iterating the construction $\mathfrak{P}_{\geq 1}$:
$$e^{e^z-1} - 1, \quad e^{e^{e^z-1}-1} - 1, \quad e^{e^{e^{e^z-1}-1}-1} - 1,$$
where, e.g., the number sequence $(1, 3, 12, 60, 358, \ldots)$ associated to the second case is A000258 of Sloane's *EIS* [58]. Related structures are of interest in finite model theory; see [68] for an introduction.
- Admissibility also covers generating functions of the type $e^{P(z)}$, with P a polynomial with nonnegative coefficients. This includes permutations with sizes of cycles constrained to be in a finite set Ω, for instance involutions

($\mathcal{I} = \mathfrak{P}(\mathfrak{C}_{1,2}(\mathcal{Z}))$, the solutions of $\sigma^d = Id$ in the symmetric group, and permutations whose longest cycle is at most some fixed value m.

The conditions of Theorem 3 are not satisfied by words without long runs (Example 1), surjections (Example 6, observe the lack of concentration on Figure 1, middle), and permutations (Example 7), although they fail by little, since the mean and standard deviation, $\nu_1(x)$ and $\sigma(x)$, happen to be of the same order of magnitude. They fail more dramatically for binary trees (Example 2 and Figure 1, bottom), secondary structures (Example 3), and noncrossing graphs (Example 4), where the ratio $\sigma(x)/\nu_1(x)$ now tends to infinity, in which case sizes produced by Boltzmann models exhibit a high dispersion. As discussed in the next two subsections and in Section 7, such situations can however be dealt with.

6.2. **Singularity types and rejection samplers.** It is possible to discuss at a fair level of generality cases where rejection sampling is efficient. The discussion is fundamentally based on the types of singularities that the generating functions exhibit. This is an otherwise well-researched topic as it is central to asymptotic enumeration [26, 28, 52].

Definition 2. *A function $f(z)$ analytic at 0 and a with finite radius of analyticity $\rho > 0$ is said to be* Δ–singular *if it satisfies the two conditions:*

(i) The function admits ρ as its only singularity on $|z| = \rho$ and it is continuable in a domain

$$\Delta(r,\theta) = \left\{ z \;\middle|\; z \neq \rho,\ |z| < r,\ \arg(z-\rho) \notin (-\theta,\theta) \right\},$$

for some $r > \rho$ and some θ satisfying $0 < \theta < \frac{\pi}{2}$.

(ii) For z tending to ρ in the Δ domain, $f(z)$ satisfies a singular expansion of the form

$$f(z) \underset{z\to\rho}{\sim} P(z) + c_0(1-z/\rho)^{-\alpha} + o((1-z/\rho)^{-\alpha}), \qquad \alpha \in \mathbb{R}\setminus\{0,-1,-2,\ldots\},$$

where $P(z)$ is a polynomial. The quantity $-\alpha$ is called the singular exponent *of $f(z)$.*

For reasons argued in [27], all the generation functions associated with specifiable models in the sense of this article are either entire or, else, they have dominant singularities which are isolated, hence they satisfy continuation conditions similar to (i). Condition (ii) is also granted in a large number of cases. Here, words without long runs, surjections, and permutations (Examples 1, 6, and 7) have generating functions with a polar singularity, corresponding to the singular exponent -1. Trees, secondary structures, and noncrossing graphs (Example 2, 3, and 4), which are recursively defined have singular exponent $\frac{1}{2}$; see [24, 49] and Section 8 below. Many properties go along with the conditions of Definition 2. Most notably, the counting sequence associated with a generating function $f(z)$ that is Δ-singular systematically obeys an asymptotic law:

$$[z^n]f(z) \sim \frac{c_0}{\Gamma(\alpha)}\rho^{-n}n^{\alpha-1}, \qquad (n\to\infty). \tag{20}$$

(This results from the singularity analysis theory exposed in [26, 28, 52].)

Returning to random generation, one has:

Theorem 5. *Let $\mathcal{C}$ be a combinatorial class such that its generating function is Δ-singular with an exponent $-\alpha < 0$. Then the rejection sampler $\mu C(x_n; n, \varepsilon)$ corresponding to a fixed tolerance $\varepsilon > 0$ succeeds in a number of trials whose expected value is asymptotic to the constant*

$$\frac{1}{\xi_\alpha(\varepsilon)}, \quad \textit{where} \quad \xi_\alpha(\varepsilon) = \frac{\alpha^\alpha}{\Gamma(\alpha)} \int_{-\varepsilon}^{\varepsilon} (1+s)^{\alpha-1} e^{-\alpha(1+s)}\, ds.$$

If $\mathcal{C}$ is specifiable, approximate-size Boltzmann sampling based on $\mu C(x_n; n, \varepsilon)$ has cost that is $O(n)$; exact-size sampling has cost $O(n^2)$.

Here is a table of numerical values of the expected number of trials $(1/\xi_{\alpha(\varepsilon)})$ for various values of the singular exponent $-\alpha$ and tolerance ε:

(21)

	$\varepsilon = 0.2$	$\varepsilon = 0.1$	$\varepsilon = 0.05$	$\varepsilon = 0.01$
$-\alpha = -2$	4.619	9.236	18.47	92.36
$-\alpha = -\frac{3}{2}$	5.387	10.80	21.61	108.0
$-\alpha = -1$	6.750	13.56	27.17	135.9
$-\alpha = -\frac{1}{2}$	9.236	20.61	41.30	206.6

For instance a tolerance of $\pm 10\%$ is likely to necessitate about 10 trials when $-\alpha$ is -2 or $-\frac{3}{2}$, while this number doubles for the singular exponent $-\frac{1}{2}$.

Proof. The rejection sampler used with the value x has a probability of success in one trial equal to

$$\mathbb{P}_x(|N/n - 1| \leq \varepsilon),$$

which is to be estimated. The inverse of this quantity gives the expected number of trials.

Functions that are Δ-singular are closed under differentiation, since, by elementary complex analysis, asymptotic expansions valid in sectors can be subjected to differentiation [54, p. 9]. Consequently, one has

$$\nu_1(x) \underset{x\to\rho^-}{\sim} \frac{\alpha x/\rho}{1 - x/\rho} \to \infty,$$

which verifies the mean value condition, whereas a similar calculation shows $\sigma(x)$ to be of the same order as $\nu_1(x)$ and the variance condition is not satisfied. The strong form of coefficient estimates in (20) then entails

$$\mathbb{P}_x(N = m) \sim \frac{1}{\Gamma(\alpha)} \frac{m^{\alpha-1}|x/\rho|^m}{(1 - x/\rho)^{-\alpha}}, \tag{22}$$

for $x \to \rho^-$ and $m \to \infty$.

Tune now the rejection sampler at the value $x = x_n$, so that $\nu_1(x_n) = n$. One has

$$x_n \sim \rho\left(1 - \frac{\alpha}{n}\right).$$

Then, setting $m = t\nu_1(x_n) = tn$ transforms the estimate (22) into

$$\begin{aligned} \mathbb{P}_x(N = \lfloor tn \rfloor) &\sim \frac{1}{\Gamma(\alpha)} \frac{t^{\alpha-1} e^{tn\log(1-(\alpha/n))}}{\alpha^{-\alpha} n} \\ &\sim \frac{1}{n\Gamma(\alpha)} \alpha^\alpha t^{\alpha-1} e^{-\alpha t}, \end{aligned} \tag{23}$$

uniformly for t in a compact subinterval of $(0, \infty)$. This is exactly a *local limit law* for Boltzmann sizes in the form of a Gamma distribution [21, p. 47].

Cumulating the estimates in the formula above, one finds (by Euler-Maclaurin summation),

$$\mathbb{P}_{x_n}(|N/n-1|\le\varepsilon)\sim\frac{\alpha^\alpha}{\Gamma(\alpha)}\int_{-\varepsilon}^{\varepsilon}(1+s)^{\alpha-1}e^{-\alpha(1+s)}ds \tag{24}$$

which gives the value $\xi_\alpha(\varepsilon)$ of the statement. Linearity for the cumulated size then follows from the moderate dispersion of sizes induced by the relation $\sigma(x)=\Theta(\nu_1(x))$.

The argument adapts when ε is allowed to tend to 0. In this case, as seen directly from (23), the success probability of a single trial is asymptotic to

$$2\frac{(\alpha e)^\alpha}{\Gamma(\alpha)}\varepsilon,$$

with the inverse of this quantity giving the mean number of trials. In particular, if the target size lies in a fixed-width window around n ($\varepsilon=O(1/n)$), which covers exact-size random sampling, then a random generation necessitates $O(n)$ trials, corresponding to an overall complexity that is $O(n^2)$ under the real-arithmetic model. ∎

Given the polar singularity involved, Theorem 5 applies directly to words without long runs (Ex. 1), surjections (Ex. 6), and cycles-in-permutations (Ex. 7).

EXAMPLE 9. *Mappings with degree constraints.* By a mapping of size n is meant here a function from $[1,n]$ into $[1,n]$. (Obviously, there are n^n of these.) We fix a finite set Ω and restrict attention to degree-constrained mappings f such that for each x in the domain, the cardinality of $f^{(-1)}(x)$ lies in Ω. (In the combinatorics literature, such mappings are surveyed in [2, 25].) For instance, in a finite field, a non-zero element has either 0 or 2 predecessors under the mapping $f;x\mapsto x^2$, so that (neglecting one exceptional value) a quadratic function may be regarded as an element of the set of mappings constrained by $\Omega=\{0,2\}$. Mappings are of interest in computational number theory as well as in cryptography [55], and the eighth Fermat number, $F_8=2^{2^8}+1$ was first factored by Brent and Pollard [5] in 1981 by means of an algorithm that precisely exploits statistical properties of degree-constrained mappings.

As is well known, a mapping can be represented as a directed graph (Figure 8) where each vertex has outdegree equal to 1, while, by the degree constraint, indegrees must lie in Ω. Then the graph of a mapping is made of components, where each component is made of a unique cycle on which trees are grafted (see, e.g., [4] for this classical construction). With $\mathfrak{P}_\Omega$ representing the set construction with a number of elements constrained to lie in Ω, the class $\mathcal{M}$ of Ω–constrained mappings is

$$\mathcal{M}=\mathfrak{P}(\mathfrak{C}(\mathcal{U})),\quad \mathcal{U}=\mathcal{Z}\star\mathfrak{P}_{\Omega-1}(\mathcal{T}),\quad \mathcal{T}=\mathcal{Z}\star\mathfrak{P}_{\Omega}(\mathcal{T}).$$

There $\mathcal{T}$ is the class of rooted labelled trees with outdegrees in Ω, $\mathcal{U}$ is the class of trees grafted on a cycle, which are such that their root degree must lie in $\Omega-1$.

Let $\phi(y):=\sum_{\omega\in\Omega}y^\omega/\omega!$. The EGF of trees, T, is implicitly defined by $T=z\phi(T)$ and one has $U=z\phi'(T)$. It has been first established by Meir and Moon [49] that the EGF $T(z)$ has systematically a singularity of the square-root type (corresponding to "failure" in the implicit function theorem, see also Lemma 3 below). Precisely, one has $T(z)\sim\tau-c\sqrt{1-z/\rho}$ as $z\to\rho$, where $\rho\equiv\rho_T$ is given by

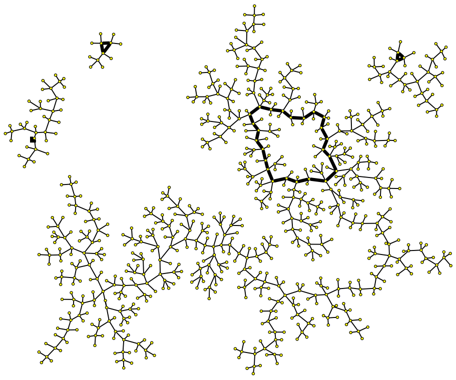


FIGURE 8. A random ternary map ($\Omega = \{0, 3\}$) of size 846 produced by Boltzmann sampling.

$\rho = \tau/\phi(\tau)$ and τ is the positive root of $\phi(\tau) - \tau\phi'(\tau) = 0$. There results that the EGF of constrained mappings satisfies as $z \to \rho$,

$$M(z) \sim \frac{1}{1 - \rho\phi'(\tau - c\sqrt{1 - z/\rho})} \sim \frac{d}{\sqrt{1 - z/\rho}},$$

for some $d > 0$. In view of this last expansion, Theorem 5 directly applies. Approximate-size random generation of Ω-constrained mappings is thus achievable in linear time. $\square$

6.3. **The pointing operator.** In this section we further enlarge the types of structures amenable to fast Boltzmann sampling. As a byproduct, we are able to lift the restriction $-\alpha < 0$ in Theorem 5, thus bringing in its scope trees, secondary structures, and noncrossing graphs (Examples 2, 3, and 4) whose singularity is known [24, 49] to be of the square–root type, i.e., $\alpha = \frac{1}{2}$.

Given a combinatorial class $\mathcal{C}$, we define the class

$$\mathcal{C}^\bullet \;=\; \{(\gamma, i) \mid \gamma \in \mathcal{C},\, i \in \{1, \ldots, |\gamma|\}\}, \quad \text{equivalently}, \quad \mathcal{C}_n^\bullet \simeq \mathcal{C}_n \times \{1, \ldots, n\},$$

of *pointed* objects. Pointing is for instance studied systematically in [4, Sec. 2.1]. Objects in $\mathcal{C}^\bullet$ may be viewed as standard objects of $\mathcal{C}$ with one of the atoms distinguished by the mark "$\bullet$". From the definition, one has $|\mathcal{C}_n^\bullet| = n|\mathcal{C}_n|$, and the GF of the class $\mathcal{C}^\bullet$ is

$$C^\bullet(z) \;=\; z\frac{d}{dz}C(z),$$

regardless of the type of $C(x)$ (ordinary or exponential). Pointing can then be viewed as a combinatorial lifting of the usual operation of taking derivatives in elementary calculus. Since any non-pointed object of $\mathcal{C}$ gives rise to exactly n pointed objects, random sampling can be equally well be performed on $\mathcal{C}_n$ or $\mathcal{C}_n^\bullet$: it suffices to "forget" the pointer in an object produced by a sampler of $C_n^\bullet$ to obtain an object of $\mathcal{C}_n$. (Only the distributions of sizes under ΓC and $\Gamma C^\bullet$ are different.)

The pointing operator $\bullet$ is related to an operator studied systematically by Greene [32] (his "box" operation) and it plays a central rôle in the recursive method (where it has been used under the name of "Theta operator"). For Boltzmann sampling, pointing can be used in conjunction with the previously defined operators $+, \times$ and $\star, \mathfrak{S}, \mathfrak{P}, \mathfrak{C}$ in either the labelled or unlabelled universe.

Lemma 2. *Let $\mathcal{C}$ be a specifiable unlabelled or labelled class (in the sense of Theorem 1 or 2). Then the class $\mathcal{C}^\bullet$ is also specifiable, i.e., it admits a specification without the pointing operator $\bullet$.*

Proof. First, for a finite class $\mathcal{C}$, the class $\mathcal{C}^\bullet$ is also finite and can be represented (and sampled) explicitly. Next, the pointing operator admits composition rules with all the other operators; in the labelled case, one has

$$(25)\qquad \left\{\begin{array}{lclclcl} (\mathcal{A}+\mathcal{B})^\bullet & = & \mathcal{A}^\bullet+\mathcal{B}^\bullet, & & (\mathcal{A}\star\mathcal{B})^\bullet & = & \mathcal{A}^\bullet\star\mathcal{B}+\mathcal{A}\star\mathcal{B}^\bullet, \\ (\mathfrak{S}\mathcal{A})^\bullet & = & \mathfrak{S}\mathcal{A}\star\mathcal{A}^\bullet\star\mathfrak{S}\mathcal{A}, & & (\mathfrak{C}\mathcal{A})^\bullet & = & \mathcal{A}^\bullet\star\mathfrak{S}\mathcal{A}, \\ (\mathfrak{P}\mathcal{A})^\bullet & = & \mathcal{A}^\bullet\star\mathfrak{P}\mathcal{A}. & & & & \end{array}\right.$$

In the unlabelled case, the first three rules apply, upon changing the labelled product "$\star$" into the cartesian product "$\times$". These rules are a combinatorial analogue of the usual differentiation rules, and have a simple interpretation: e.g., pointing at a sequence $((\mathfrak{S}\mathcal{A})^\bullet)$ implies pointing at a component $(\mathcal{A}^\bullet)$, which breaks the chain and individuates a left $(\mathfrak{S}\mathcal{A})$ and a right $(\mathfrak{S}\mathcal{A})$ subsequence.

Consider now a specification of the class $\mathcal{C}=\mathcal{F}_1$ in the form of a system,

$$\mathcal{S} \;=\; \{\mathcal{F}_i \;=\; \Phi_i(\mathcal{Z};\mathcal{F}_1,\dots,\mathcal{F}_m),\; i=1,\dots,m\},$$

where $\mathcal{F}_i$ are auxiliary classes and the Φ_i are functional terms involving finite classes and the standard operators (without pointing). Then, one can build a specification of the class $\mathcal{C}^\bullet$ in the form of a derived system,

$$\mathcal{S}'=\mathcal{S}\,\cup\,\{\mathcal{F}_i^\bullet \;=\; \Psi_i(\mathcal{Z};\mathcal{F}_1,\dots,\mathcal{F}_m,\mathcal{F}_1^\bullet,\dots,\mathcal{F}_m^\bullet),\; i=1,\dots,m\},$$

where the functionals Ψ_i do not involve the pointing operator "$\bullet$": Ψ_i is obtained from $\Phi_i^\bullet$ by application of the derivation rules until the pointing operator is applied to variables only. In the derived specification, each $\mathcal{F}_i^\bullet$ is treated as a new variable, thereby leading to a complete elimination of the pointing operator within constructions. ∎

Our interest for pointing lies in the following two observations.

- If a class $\mathcal{C}$ has a generating function $C(z)$ that is Δ-analytic with exponent $-\alpha$, then the generating function $zC'(z)$ of the class $\mathcal{C}^\bullet$ is also Δ-analytic and has an exponent $-\alpha-1$, which is smaller.
- Uniform sampling in $\mathcal{C}_n$ is equivalent to uniform sampling in $\mathcal{C}_n^\bullet$. As a consequence, the sampler $\mu\mathcal{C}^\bullet(x;n,\varepsilon)$ is a correct approximate-size sampler for the class $\mathcal{C}$ (upon removing the mark).

Let $\mu\mathcal{C}^{\bullet k}(x;n,\varepsilon)$ denote the rejection sampler of the class derived from $\mathcal{C}$ by k successive applications of the pointing operator. The last two observations immediately lead to an extension of Theorem 5:

Theorem 6. *Let $\mathcal{C}$ be a combinatorial class such that its generating function is Δ-singular with* any *exponent $-\alpha \neq \{0,1,\dots\}$. Let $\alpha_+=\max(0,\lceil -\alpha\rceil)$ the integral positive part of $-\alpha$, and $\alpha_0=\alpha+\alpha_+$ its fractional part. Then the rejection sampler $\mu\mathcal{C}^{\bullet\alpha_+}(x_n;n,\varepsilon)$ corresponding to a fixed tolerance $\varepsilon>0$ succeeds in a number of trials whose mean is asymptotic to the constant $\frac{1}{\xi_{\alpha_0}(\varepsilon)}$. In particular, if $\mathcal{C}$ is specifiable, the total cost of the rejection sampler $\mu\mathcal{C}^{\bullet\alpha_+}(x_n;n,\varepsilon)$ is $O(n)$ on average.*

As an illustration of Theorem 6, we examine the internal workings of the algorithm that results for the class $\mathcal{B}$ of binary trees taken here for convenience as

$$\mathcal{B} = \mathcal{Z} + (\mathcal{B} \times \mathcal{B}),$$

so that only external nodes contribute to size. The pointed class satisfies

$$\mathcal{B}^\bullet = \mathcal{Z}^\bullet + (\mathcal{B}^\bullet \times \mathcal{B}) + (\mathcal{B} \times \mathcal{B}^\bullet),$$

which completely defines it in terms of $\mathcal{B}$ and itself. Accordingly, the Boltzmann samplers for $\mathcal{B}$ and $\mathcal{B}^\bullet$ are defined by the system of simultaneous equations

$$\begin{cases} \Gamma B(x) &= \big(\mathrm{Bern}\,(p_0) \longrightarrow \mathcal{Z} \;\big|\; (\Gamma B(x); \Gamma B(x))\big) \\ \Gamma B^\bullet(x) &= \big(\mathrm{Bern}\,(p_1, p_2) \longrightarrow \mathcal{Z}^\bullet \;\big|\; (\Gamma B^\bullet(x); \Gamma B(x)) \;\big|\; (\Gamma B(x); \Gamma B^\bullet(x))\big) \end{cases}$$

where

$$p_0 = \frac{2x}{1-\sqrt{1-4x}}, \quad p_1 = \sqrt{1-4x}, \quad p_2 = \frac{1}{2} - \frac{1}{2}\sqrt{1-4x},$$

and the notation (9) for probabilistic switches is employed.

Random generation of a tree of size near n is achieved by a call to $\Gamma B^\bullet(x_n)$. For large n, the quantity x_n is very close to the critical value $\rho = \frac{1}{4}$. Thus, $\Gamma B^\bullet$ generates a terminal node with a small probability (since $p_1 \approx 0$), and, with high probability, $\Gamma B^\bullet(x_n)$ triggers a long sequence of calls to ΓB, which itself produces each time a near-critical tree (since $p_0 \approx \frac{1}{2}$) . In particular, the "danger" of generating small trees is automatically controlled by $\Gamma B^\bullet$. Observe that a sampler formally equivalent to $\Gamma B^\bullet(x)$ (by recursion removal) is then as follows: generate a long random branch (with randomly chosen $(\frac{1}{2}, \frac{1}{2})$ left or right branchings) and attach to it a collection of (near) critical trees[5]. For instance, here are the sizes observed in runs of 20 calls, one relative to ΓB equipped with the value $x_{500} = 0.2499997495$, the other to $\Gamma B^\bullet$ equipped with $x'_{500} = 0.2497497497$:

$$2, 1, 4, 5, 4, 1, 1, 1, 1, 1, 1, 1, 56, 1, 1, 7, 2, 1, 2, 2$$
$$831, 6, 76, 120, 1, 532, 15, 7, 11, 68, 99, 45, 1176, 12, 94, 81, 784, 3393, 21, 493.$$

(See also (15) for more extensive data that are similar to the first line.) While the parameters are chosen in each case such that the resulting object has expected size $n = 500$, it is clear that the $\Gamma B^\bullet$ sampler gets a better shot at the target.

Pointing also constitutes a valuable optimization whenever structures are driven by a cycle construction. Define a function f to be logarithmic if it is continuable in a Δ–domain and satisfies

$$f(z) = c \log \frac{1}{1 - z/\rho} + O(1), \qquad z \to \rho.$$

This may somehow be regarded as the limit case $\alpha \to 0$ of a singular exponent $-\alpha$. As the table (21) suggests, the efficiency of rejection deteriorates in this case: singularity analysis may be used to verify that $\sigma(x_n) = n\sqrt{\log n}$, so that approximate-size is of superlinear complexity, namely $O(n\sqrt{\log n})$. This problem is readily fixed by pointing. If $\mathcal{C} = \mathfrak{C}(\mathcal{A})$, then the transformation rules of (25) imply that we can alternatively generate a sequence, which is amenable to straight rejection sampling in linear time, since its generating function now has a polar-like singularity (with exponent $-\alpha = -1$). For instance, the class $\mathcal{K}$ of connected mappings is defined by

$$\{\mathcal{K} = \mathfrak{C}(\mathcal{T}), \quad \mathcal{T} = \mathcal{Z} \star \mathfrak{P}(\mathcal{T})\}.$$

[5]This construction is akin to the "size-biased" Galton–Watson process exposed in [47]. It is interesting to note that we are here led naturally to it by a systematic use of formal transformations.

The derived specification for $\mathcal{K}^\bullet$ is then

$$\{\mathcal{K}^\bullet = \mathcal{T}^\bullet \star \mathfrak{S}(\mathcal{T}),\ \mathcal{T} = \mathcal{Z} \star \mathfrak{P}(\mathcal{T}),\ \mathcal{T}^\bullet = \mathcal{Z}^\bullet \star \mathfrak{P}(\mathcal{T}) + \mathcal{Z} \star \mathfrak{P}(\mathcal{T}) \star \mathcal{T}^\bullet\},$$

with nonterminals $\mathcal{K}^\bullet, \mathcal{T}, \mathcal{T}^\bullet$. The generator $\Gamma\mathcal{K}^\bullet$ then achieves linear time sampling for any fixed tolerance $\varepsilon > 0$. (Figure 8 has been similarly produced by pointing.)

This technique applies to plane trees and variants thereof (Example 2), secondary structures (Example 3), and noncrossing graphs (Example 4). It also applies to all the simple families of labelled nonplane trees, $\mathcal{T} = \mathcal{Z} \star \mathfrak{P}_\Omega(\mathcal{T})$ defined by restrictions on node degrees (Example 9). In all these cases, linear-time approximate-size sampling is granted by Theorem 6.

7. Singular Boltzmann samplers.

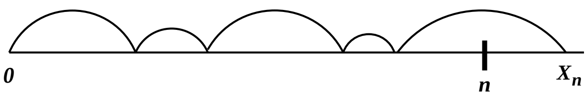
We now discuss two *infinite* categories of models, where it is possible to place oneself right at the singularity $x = \rho_C$ in order to develop rejection samplers from Boltzmann models. These "singular" rejection generators are freed from the necessity to adapt the control variable x to the target size n, thus making available implementations that only need a *fixed* set of constants to be determined once and for all, this independently of the value of n.

7.1. **Singular samplers for sequences.** The first type of singular generator we present is dedicated to the sequence construction: define a sequence construction to be *supercritical* if $\mathcal{C} = \mathfrak{S}(\mathcal{A})$ and $\rho_A > \rho_C$ (so that $A(\rho_A^-) > 1$)). Put otherwise, the generating function of components $A(x)$ should cross the value 1 before it becomes singular. The generating function of $\mathcal{C}$ and $\mathcal{A}$ satisfy $C(z) = 1/(1 - A(z))$, so that the supercriticality condition implies that $A(\rho_C) = 1$ and the (dominant) singularity ρ_C of $C(x)$ is a pole. (This notion of supercriticality is borrowed from Soria [59] who showed it to be determinant in the probabilistic properties of sequences.)

Literally taken, the Boltzmann sampler ΓC of Section 3 taken with $x = \rho_C$ loops forever and generates objects of infinite size, as it produces a number of components equal to a "Geom(1)". This prevents us from using the rejection algorithm $\mu\mathcal{C}(x; n, \varepsilon)$ with $x = \rho$. However, one may adapt the idea by halting execution as soon as the target size has been attained. Precisely, the early-interrupt *singular sequence sampler* is defined as follows:

function $\sigma C(\rho; n)$; {Early-interrupt singular sequence sampler}
$i := 0$; repeat $i := i + 1$; $\gamma_i := \Gamma A(\rho)$ until $|(\gamma_1, \ldots, \gamma_i)| > n$;
return$((\gamma_1, \ldots, \gamma_i))$; end.

The principle of the algorithm can be depicted as "leapfrogging" over n:



The *singular early-interrupt* sampler determined by the choice $x = \rho_C$ has excellent probabilistic and complexity-theoretic characteristics summarized in the following statement. There, we assume without loss of generality that $A(z)$ is aperiodic in the sense that the quantity $d := \gcd\{n \mid A_n \neq 0\}$ satisfies $d = 1$. (If $d \geq 2$, a linear change of the size functions brings us back to the aperiodic case.)

Theorem 7. *Consider a sequence construction, $\mathcal{C} = \mathfrak{S}(\mathcal{A})$ that is supercritical and aperiodic. Then the early-interrupt singular sequence generator, $\sigma C(\rho_C; n)$ is a valid sampler for $\mathcal{C}$. It produces an object of size $n + O(1)$ in one trial with* *high probability. For a specifiable class $\mathcal{A}$, exact-size random generation in $\mathcal{C}$ is achievable from this generator by rejection in expected time $O(n)$.*

Proof. Let X_n denote the random variable giving the size of the output of the early-interrupt singular sequence generator with tarket size n. The analysis of X_n can be treated by classical renewal theory [20, Sec. XIII.10], but we opt for a direct approach based on generating functions, which integrates smoothly within our general formalism.

The bivariate (probability) generating function with variable z marking the target size n and variable u marking the size X_n of the actually generated object is

$$F(z,u) := \sum_{n\geq 1}\sum_{m\geq n} \mathbb{P}(X_n = m)\, z^n u^m.$$

A trial decomposes into a sequence of samples of $\Gamma\mathcal{A}(\rho)$ ending by a sample that brings the total over n, which implies

$$F(z,u) = \frac{1}{1-A(\rho z u)}\mathcal{L}[A(\rho z u)] = \frac{z}{1-z}\frac{A(\rho u)-A(\rho z u)}{1-A(\rho z u)}.$$

There $\mathcal{L}[f(z)] := z(f(1)-f(z))/(1-z)$ is a linear operator, and, e.g.,

$$\mathcal{L}[\frac{1}{1-zu}] = z(u+u^2+\cdots) + z^2(u^2+u^3+\cdots) + z^3(u^3+u^4+\cdots)+\cdots,$$

so that all powers of the form $z^n u^\ell$ with $\ell \geq n$ are produced.

One checks that $F(z,1) = z/(1-z)$, as should be. Next the expected size $\mathbb{E}(X_n)$ of the output is given by the coefficient of z^n in

$$\begin{aligned}
\frac{\partial}{\partial u}F(z,u)\big|_{u=1} &= \frac{z}{1-z}\frac{\rho A'(\rho)}{1-A(\rho z)} \\
&= \frac{z}{(1-z)^2} + \frac{\rho A''(\rho)}{2A'(\rho)}\cdot\frac{z}{1-z} + O(1) \qquad (z\to 1).
\end{aligned}$$

This expansion at the polar singularity 1 then yields the expected “overshoot”:

$$\mathbb{E}(X_n - n) \;=\; [z^n]\frac{\partial}{\partial u}F(z,u)\big|_{u=1} \;-\; n \;=\; \frac{\rho A''(\rho)}{2A'(\rho)} + O(1/n).$$

The second moment of the expected size of the output is similarly obtained via two successive differentiations. A simple computation then shows the variance of the overshoot to satisfy

$$\mathbb{E}((X_n-n)^2) - \mathbb{E}(X_n-n)^2 \;=\; O(1).$$

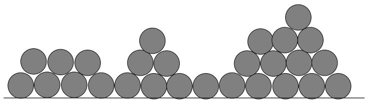
As a matter of fact, the discrete distribution of the overshoot is described by

$$\begin{aligned}
\mathbb{P}(X_n - n \,=\, m) &= [z^n u^{n+m}]F(z,u) \;=\; [z^n u^m]\frac{z}{u-z}\left(1-\frac{1-A(\rho u)}{1-A(\rho z)}\right), \\
&= [z^{n+m}]\frac{1}{1-A(\rho z)} - \sum_{\ell=0}^{m-1}[z^{n+\ell}]\frac{1}{1-A(\rho z)}[u^{m-\ell}]A(\rho u), \\
&= \left(\frac{1}{\rho A'(\rho)} + O(1/n)\right)\left(1-\sum_{\ell=0}^{m-1}\mathbb{P}(N\,=\,\ell)\right) \\
&= \frac{\mathbb{P}(N\geq m)}{\mathbb{E}(N)} + O(1/n).
\end{aligned}$$

where N denotes the random size of an element of $\mathcal{A}$ under the Boltzmann model of parameter ρ and the two last estimates hold for $n \to \infty$ uniformly in m. The distribution of N has exponential tails (since $\rho \equiv \rho_C$ lies strictly within the disc of convergence of $A(z)$), and thus the probability of a large overshoot decays geometrically fast. This proves that exact size n is attained in $O(1)$ trials. ∎

This theorem applies to "cores"of words without long runs (Equation (14) from Example 1) and it can be adapted to yield a generator of the full set $\mathcal{R}$. It applies to surjections (Example 6), for which exact-size generation becomes possible in linear time. It also provides a global setting for a variety of *ad hoc* algorithms developed by Louchard [43, 44, 46] in the context of efficient generation of certain types (directed, convex) of random planar diagrams known as "animals" and "polyominos".

EXAMPLE 10. *Coin fountains ($\mathcal{O}$).* A fountain is formed by starting with a row of coins, then stacking additional coins on top so that each new coin touches two in the previous row, for instance,



These configurations have been enumerated by Odlyzko and Wilf [53] and the counting sequence starts as (A005169 of [58])

$$1, 1, 1, 2, 3, 5, 9, 15, 26, 45, 78, 135, 234, 406, 704, \ldots$$

They correspond to Dyck paths (equivalently, Bernoulli excursions) taken according to area but disregarding length. A decomposition by slices taken at an angle of $\frac{2}{3}\pi$ (on the example, this gives 1,2,2,2,1,2,3,1,1,2,3,3,4) is then expressed by an infinite specification (not *a priori* covered by the standard paradigm):

$$\mathfrak{S}(\mathcal{Z}\mathfrak{S}(\mathcal{Z}^2\mathfrak{S}(\mathcal{Z}^3\mathfrak{S}(\cdots)))).$$

The OGF is consequently given by the continued fraction (see also [23]),

$$O(z) = \cfrac{1}{1 - \cfrac{z}{1 - \cfrac{z^2}{1 - \cfrac{z^3}{\cdots}}}}.$$

At top level, the singular Boltzmann sampler of Theorem 7 applies (write $\mathcal{O} = \mathfrak{S}(\mathcal{Q})$ and $O(z) = (1 - Q(z))^{-1}$), this even though $\mathcal{O}$ is not finitely specifiable. The root ρ of $Q(z) = 1$ is easily found to 50D,

$$\rho \doteq 0.57614876914275660229786857371993878235472466311 89,$$

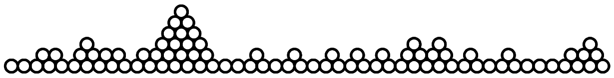
see [53] for a transcendental equation satisfied by ρ that involves the q–exponential. The objects of $\mathcal{Q}$ needed are then with high probability of size at most $O(\log n)$ (by general properties of largest components in sequences [31]), so that they can be generated by whichever subexponential method is convenient (e.g., MAPLE's Combstruct) to the effect that the overall (theoretical and practical) complexity remains $O(n)$.

Precisely, the implementation runs like this. First define a family of finitely specifiable approximants to $\mathcal{Q}$, as follows:

$$\mathcal{Q}^{[j]} := \mathcal{Z}\mathfrak{S}(\mathcal{Z}^2\mathfrak{S}(\mathcal{Z}^3\mathfrak{S}(\cdots\mathcal{Z}^{j-1}\mathfrak{S}(\mathcal{Z}^j)\cdots))).$$

At any given time, the program operates with the class $Q^{[d]}$ of depth d: $Q^{[d]}(z)$ and $Q(z)$ coincide till terms of order $\nu(d) = \binom{d+1}{2} - 1$. The corresponding counts till $\nu(d)$ are assumed to be available, together with the corresponding exact-size samplers for $\mathcal{Q}^{[d]}$. (It is proves especially convenient here to appeal to algorithms based on the recursive method as provided by Combstruct.) In this way, one "knows" how to sample from Q till size $\nu(d)$, and from knowledge of the precise value of ρ, one also "knows" whenever a $\mathcal{Q}$ component of size larger than $\nu(d)$ might be required. (If so, adaptively increase the value of d and resume execution.) For instance, taking $d = 4$ (with $\nu = 9$) already suffices in 92% of the cases to produce an element of ΓQ, while $d = 20$ (and $\nu = 104$) suffices with probability about $1 - 2 \cdot 10^{-19}$ and is thus likely to cater for all simulation needs one might ever have.

The resulting implementation constants are reasonably *low*, so that random generation in the range of millions becomes feasible thanks to the singular Boltzmann generator. Here is for instance a fragment of a random fountain of size 100,004 ($n = 10^5$) obtained in this way (in only about a trillion clock cycles under MAPLE):



Dutour et al. [19] have previously employed an adaptation of the recursive method, but it is limited to sizes perhaps in the order of a few hundreds. $\square$

EXAMPLE 11. *Weighted Dyck paths and adsorbing staircase walks.* In [48], Martin and Randall examine (under the name of adsorbing walks) the generation of Dyck paths of length $2n$, where a path receives a weight proportional to λ^k if it hits the horizontal axis k times. Their Markov chain based algorithm has a high polynomial time complexity, perhaps as much as $O(n^{10})$, if not beyond. In contrast, for $\lambda > 2$, a Boltzmann sampler based on supercritical sequences has a complexity that is $O(n)$, this even when exact-size random generation is required. Precisely, let $\mathcal{D}$ be the class of Dyck paths defined by the grammar $\mathcal{D} = \mathbf{1} + \nearrow \mathcal{D} \searrow \mathcal{D}$ with OGF $D(z) = (1 - \sqrt{1-4z})/(2z)$ (with z marking size taken here to be half-length). One needs to generate objects from the weighted class $\mathcal{E} := \mathfrak{S}(\nearrow \mathcal{D} \searrow)$, viewed as weighted sequences of "arches" with OGF $(1 - z\lambda D(z))^{-1}$, where the coefficient

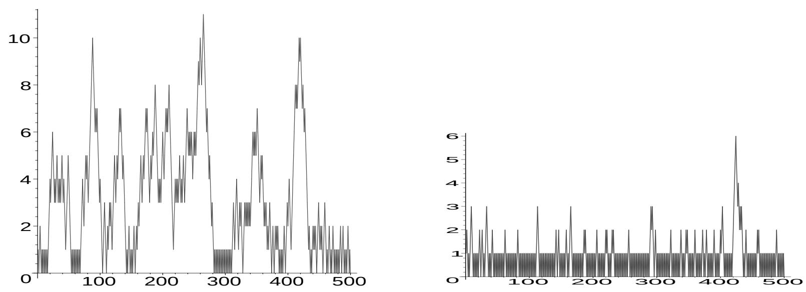


FIGURE 9. Weighted Dyck paths of length 500 corresponding to $\lambda = 2.1$ (left) and $\lambda = 3.1$ (right).

λ takes the proper weighting into account. The sequence is supercritical as soon as $\lambda > 2$, and the singular value of the Boltzmann parameter is found to be at $\rho = (\lambda - 1)/\lambda^2$. Then, the linear time generator is, for $\lambda > 2$:

let $\rho := \frac{\lambda-1}{\lambda^2}$, $D_k = \frac{1}{k+1}\binom{2k}{k}$;
repeat $S := 0$; repeat
generate K according to the distribution $\{\frac{\lambda-1}{\lambda} D_k \rho^k\}_{k=0}^{\infty}$;
$S := S + 2K + 2$; draw at random from $\nearrow \mathcal{D}_K \searrow$; {e.g., in linear time}
until $S \geq 2n$; until $S = 2n$.

There, the last successful run should be returned. (The case where $\lambda \leq 2$ is easily treated in linear time by direct combinatorics.) Figure 9 displays two such paths of length 500 (higher values of λ increase the number of contacts). $\square$

The book by van Rensburg [66] describes models similar to the last two ones (in the context of critical phenomena in polymers and vesicles), a number of which are amenable to efficient Boltzmann sampling as they correspond to combinatorial classes that are specifiable.

7.2. **Singular samplers for recursive structures.** Recursive structures tend to conform to a universal complex-analytic pattern corresponding to a square-root singularity, that is, a singular exponent $-\alpha = 1/2$. This specific behaviour may be exploited, resulting in another variety of singular samplers.

In the statement below, a recursive class $\mathcal{C}$ is defined as the component $\mathcal{C} = \mathcal{F}_1$ of a system of mutually dependent equations,

$$\{\mathcal{F}_1 = \Psi_1(\mathcal{Z}; \mathcal{F}_1, \ldots, \mathcal{F}_m), \ldots, \mathcal{F}_m = \Psi_m(\mathcal{Z}; \mathcal{F}_1, \ldots, \mathcal{F}_m)\}$$

where the Ψ's are *any* functional term involving any of the basic constructors previously defined ('$+$', '$\times$' or '$\star$', and $\mathfrak{S}, \mathfrak{P}, \mathfrak{C}$; pointing is not allowed here). The system is said to be *irreducible* if the dependency graph between the $\mathcal{F}_j$ is strongly connected (every nonterminal $\mathcal{F}_j$ depends on any other $\mathcal{F}_k$). A class $\mathcal{F}$ is said to be of *lattice type* if the index set of the nonzero coefficients of $F(z)$ is contained in an arithmetic progression of some ratio d, with $d \geq 2$. (The terminology is borrowed from classical probability theory.) For instance, the class of "complete" binary trees ($\mathcal{F} = \mathcal{Z} + \mathcal{Z}\mathcal{F}^2$) only has objects of size $n = 1, 3, 5, 7, \ldots$, and is consequently lattice of ratio 2. Any lattice class is equivalent to a nonlattice one, upon redefining size via a linear transformation.

Lemma 3. *Consider a combinatorial class $\mathcal{C}$ defined by a recursive specification that is irreducible and non-lattice. Then $C(z)$ has a unique dominant singularity which is algebraic and of the square-root type, that is, with singular exponent $-\alpha = 1/2$ in the notations of Section 6.2.*

Proof (sketch). The $F_j(x)$ are implicitly defined by an image system $\mathbf{F} = \Psi[\mathbf{F}]$. The Jacobian matrix of Ψ,

$$\mathbf{J}(z) := \left(\frac{\partial}{\partial F_i}\Psi_j(\mathbf{F})\right)_{i,j}$$

is at least defined near the origin. Let $\lambda(z)$ be the spectral radius of $\mathbf{J}(z)$. For small enough positive x, the matrix $\mathbf{J}(x)$ is Perron–Frobenius by irreducibility. A local analysis of the Drmota–Lalley–Woods type [16, 41, 70] based on "failure" of the implicit function theorem in its analytic version establishes the following: each F_j has a singularity at ρ which is determined as the smallest positive root of $\det \mathbf{J}(x) = 1$, and the behaviour of F_j there is of the square-root type in a Δ-domain. The non-lattice assumption implies that each F_j satisfies $|F(z)| < F(|z|)$ for any z satisfying $0 < |z| < \rho$ and $z \notin \mathbb{R}_{>0}$; by domination properties of analytic functions with positive coefficients and matrices with complex entries, this implies that $\lambda(z) < \lambda(|z|)$, whence the fact that each F_j is analytic on $|z| = \rho$, $z \neq \rho$. ∎

In view of Lemma 3, $C(z)$ is Δ-singular with an expansion of the form

$$C(x) \;=\; C(\rho) - c_0(1 - z/\rho)^{1/2} + O(1 - z/\rho), \tag{26}$$

where $C(\rho) > 0$ and $c_0 > 0$. Singularity analysis then implies that the coefficients are asymptotically given by

$$[z^n]C(z) = \frac{c_0}{2\sqrt{\pi}}\rho^{-n}n^{-3/2}\left(1 + O(n^{-1})\right). \tag{27}$$

(For details see [28, Ch. 8] and reference therein.) Consequently, the distribution of sizes at the critical value $x = \rho$ is of the form $\mathbb{P}(N = n) \propto n^{-3/2}$, which means that it has heavy tails. In particular, the expectation of size $\mathbb{E}(N)$ is infinite (this fact is well-known in the special case of critical branching processes). Such an observation precludes the use of straight-rejection Boltzmann sampling.

The idea of an early interruption discussed in the previous section may be adapted and extended. Consider in all generality a Boltzmann sampler $\Gamma C(x)$ built according to the design principles already exposed and let m be a *ceiling* (i.e., an uppperbound) imposed on the size of the required objects. It is possible to build a modification $\Gamma C^{<m}(x)$ of $\Gamma C(x)$ as follows: maintain a running count, implemented as a global counter K, of the number of atoms produced at any given time during a partial execution of sampling by $\Gamma C(x)$; the counter is regularly incremented as long as $K \leq m$ each time an atom is produced; however, as soon as K exceeds m, execution is interrupted and the "undefined" symbol $\perp$ is returned. Then, rejection can be piled on top of this sampler, which corresponds to the scheme:

function $\nu C(x; n, \varepsilon)$; {Ceiled rejection sampler}
repeat $\gamma := \Gamma C^{<m}(x; n(1+\varepsilon))$ until $(\gamma \neq \perp) \wedge (|\gamma| \geq n(1-\varepsilon))$;
return(γ); end.

This ceiling technique optimizes *any* Boltzmann sampler for *any* value of x. The choice of the singular value $x = \rho$ makes the algorithm well-behaved for recursive classes.

Theorem 8. *Let $\mathcal{C}$ be a combinatorial class given by a recursive specification that is irreducible and aperiodic. Then the singular ceiled rejection sampler $\nu\mathcal{C}(\rho; n, \varepsilon)$, corresponding to a fixed tolerance $\varepsilon > 0$ succeeds in a number of trials whose expected value grows like $n^{1/2}/\zeta(\varepsilon)$ for a positive constant $\zeta(\varepsilon)$ given by* (30) *below.*

Moreover the cumulated size T_n of the generated and rejected objects during the call of $\nu\mathcal{C}(\rho; n, \varepsilon)$ satisfies as $n \to \infty$

$$\mathbb{E}(T_n) \sim \frac{n}{\varepsilon}\left((1-\varepsilon)^{1/2} + (1+\varepsilon)^{1/2}\right) \tag{28}$$

with its variance, $\sigma^2 = \mathbb{E}(T_n^2) - \mathbb{E}(T_n)^2$, being

$$\sigma^2 \sim \mathbb{E}(T_n)^2 + \frac{n^2}{\varepsilon}\left(\frac{1}{3}(1-\varepsilon)^{3/2} + (1+\varepsilon)^{3/2}\right). \tag{29}$$

Under these conditions, approximate-size sampling and exact-size sampling are of average-case complexity respectively $O(n)$ and $O(n^2)$.

Proof. Let $C(x)$ be the generating function of $\mathcal{C}$, and let $C^{<n_1}(x)C^{>n_2}(x), C^{[n_1,n_2]}(x)$ be the generating function for the subclass of those objects with size respectively strictly less than $n_1 = (1-\varepsilon)n$, strictly greater than $n_2 = (1+\varepsilon)n$, and between n_1 and n_2. The coefficients of $C(z)$ are known from Equation (27), so that $\Gamma C(\rho)$ produces sizes according to

$$\mathbb{P}(N = k) \ \sim \ \frac{c_0}{2C(\rho)\sqrt{\pi}}\, k^{-3/2}.$$

For any $\varepsilon > 0$, the probability that a single trial (one execution of the repeat loop) of the ceiled rejection sampler $\nu C(\rho; n, \varepsilon)$ succeeds is obtained by summing over all values of k in the interval $[n(1-\varepsilon), n(1+\varepsilon)]$. This probability thus decays like $\zeta(\varepsilon)n^{-1/2}$ where

$$\zeta(\varepsilon) \ = \ \frac{c_0}{5C(\rho)\sqrt{\pi}}\,((1+\varepsilon)^{5/2} - (1-\varepsilon)^{5/2}). \tag{30}$$

The expected number of trials follows.

Next, the probability generating function of the interruptive singular Boltzmann sampler targeted at $[n_1, n_2]$ is

$$F(u) \ = \ \sum_k \mathbb{P}(T_n = k)\, u^k.$$

From the decomposition of a call to $\nu\mathcal{C}$ into a sequence of unsuccessful trials (contributing to T_n) followed by a final successful trial (not contributing to T_n),

$$F(u) = \left(1 - \frac{1}{C(\rho)}\left(C^{<n_1}(\rho u) + C^{>n_2}(\rho)u^{n_2}\right)\right)^{-1} \frac{C^{[n_1,n_2]}(\rho)}{C(\rho)}.$$

(This is the cost *in addition* to the size of the last successful output, and it is assumed that the generation of objects with size larger than n_2 is interrupted at size n_2.) The moments of the cost are then given by

$$\mathbb{E}(T_n) = \frac{\partial}{\partial u}F(u)\big|_{u=1}, \qquad \mathbb{E}(T_n^2) = \frac{(u\partial)^2}{\partial u^2}F(u)\big|_{u=1}.$$

Taking partial derivatives, then specializing to $u = 1$, and observing that $C(x) - C^{<n_1}(x) - C^{>n_2}(x) = C^{[n_1,n_2]}(x)$, we get

$$\begin{aligned}
\mathbb{E}(T_n) \ &= \ \frac{\rho C'^{<n_1}(\rho) + n_2 C^{>n_2}(\rho)}{C^{[n_1,n_2]}(\rho)}, \\
\mathbb{E}(T_n^2) \ &= \ \frac{\rho^2 C''^{<n_1}(\rho) + n_2(n_2-1)C^{>n_2}(\rho)}{C^{[n_1,n_2]}(\rho)} + 2\,\mathbb{E}(T_n)^2 \ + \ \mathbb{E}(T_n).
\end{aligned}$$

The asymptotic expression for the coefficients of $C(x)$ as given in (27) yields, by direct Euler-MacLaurin summation:

$$\begin{array}{llllll}
\rho C'^{<n_1}(\rho) & \sim & 2c_0 n_1^{1/2}, & \rho^2 C''^{<n_1}(\rho) & \sim & \frac{2c_0}{3} n_1^{3/2}, \\
C^{>n_2}(\rho) & \sim & 2c_0 n_2^{-1/2}, & C^{[n_1,n_2]}(\rho) & \sim & 2c_0\varepsilon n^{-1/2}.
\end{array} \tag{31}$$

The estimates (31) combine with the exact expressions of $\mathbb{E}(T_n)$ and $\mathbb{E}(T_n^2)$ to give the values stated in (28) and (29).

For a relative tolerance $\varepsilon = \varepsilon_n$ depending on n and tending to zero, the estimates become $\mathbb{E}(T_n) \sim \frac{2n}{\varepsilon}$ and $\sigma \sim \mathbb{E}(T_n)$, which implies the quadratic cost of exact-size sampling. ∎

The singular ceiled rejection sampler thus provides linear-time approximate-size random generation for all the simple varieties of trees of Example 2, including binary trees, unary-binary trees, 2-3 trees, and so on, for secondary structures (Example 3), and for noncrossing graphs (Example 4). In all these cases, exact-size is also achievable in quadratic time. The method does not require the pointing transformations of Section 6.3 and only necessitates a fixed number of constants, themselves independent of the target value n. The technique is akin to the "Florentine algorithm" invented by Barcucci–Pinzani–Sprugnoli [3] to generate prefixes of Motzkin words and some directed plane animals. The cost analysis given above is related to Louchard's work [45].

Note. Let $\mathcal{T}$ be a class of trees determined by restricting the degrees of nodes to lie in a finite set Ω, that is, $\mathcal{T} = \mathfrak{S}_\Omega(\mathcal{T})$ or $\mathcal{T} = \mathfrak{P}_\Omega(\mathcal{T})$, depending on whether the trees are embedded in the plane or not. The corresponding generating function satisfies $T(z) = z\phi(T(z))$ (see Example 9). For such trees, exact-size sampling can be performed in time $O(n^{3/2})$, which improves on the general bound $O(n^2)$ of Theorem 8. Indeed, in order to generate a tree of size n, it suffices to generate a Łukasiewicz code of length n, with steps in $\Omega - 1$. By Raney's conjugacy principle [42, Ch. 11] (also known as Dvoretzky and Motzkin's cycle lemma), this task itself reduces to generating at random a planar path of length n with steps in $\Omega - 1$ and with final altitude -1. When one places oneself right at the singular value ρ (for $T(z)$), the latter task is equivalent to sampling from n independent random variables, having support $\Omega - 1$ and probability generating function $\psi(z) = \phi(\rho z)/(z\phi(\rho))$, conditioned to sum to the value -1. Rejection (on the final value of the n–sum) achieves this in $O(n^{1/2})$ trials, by virtue of the local limit theorem for sums of discrete random variables. In this way, trees from any finitely generated family of trees can be sampled in total time $O(n^{3/2})$; equivalently, the technique makes it possible to sample from any branching process (with finitely supported offspring distribution) conditioned upon the size of the total progeny being n, this again in time $O(n^{3/2})$.

8. Conclusions

As shown here, combinatorial decompositions allow for random generation in low polynomial time. In particular, approximate-size random generation can often be effected in linear time, using algorithms that suitably exploit the "physics" of random combinatorial structures. Given the large number of combinatorial decompositions that have been gathered over the past two decades (see, e.g., [4, 28, 30]) we thus estimate to well over a hundred the number of classical combinatorial structures that are amenable to efficient Boltzmann sampling. In contrast with the recursive method [13, 29, 51], memory requirements are kept to a minimum since only a table of constants of size $O(1)$ is required.

For the reader's convenience, we gather in Figure 10 the best strategies that have been developed for each of the eleven pilot examples of this article. Naturally, a few of the basic cases are beaten by special-purpose combinatorial generators—this happens for permutations ($\mathcal{P}$), binary trees ($\mathcal{B}$), or mappings ($\mathcal{M}$) and Cayley trees ($\mathcal{T}$), where the counting sequences admit of a product form and specific bijections may be exploited to achieve exact-size sampling in linear time [51]. In such cases, however, the same complexity estimates continue to hold when Boltzmann sampling is applied to a large number of related classes, whereas dedicated combinatorial generators based on bijections generally break down. For instance, Boltzmann

	Structures		*Approx. size*	*Exact size*
1.	Runs	$\mathcal{R}$	$O(n)$ (reject.)	$O(n)$ (sing. seq.)
2.	Trees	$\mathcal{B}$	$O(n)$ (point.; sing. ceil.)	$O(n^2)$ (point.; sing. ceil.); $O(n^{3/2})$
3.	Secondary S.	$\mathcal{W}$	$O(n)$ (point.; sing. ceil.)	$O(n^2)$ (point.; sing. ceil.)
4.	Noncrossing G.	$\mathcal{X}$	$O(n)$ (point.; sing. ceil.)	$O(n^2)$ (point.; sing. ceil.)
5.	Set Part.	$\mathcal{S}$	$O(n)$ (reject.)	$O(n^{3/2}\sqrt{\log n})$ (reject.)
6.	Surjections	$\mathcal{Q}$	$O(n)$ (reject.)	$O(n)$ (sing. seq.)
7.	Permutations	$\mathcal{P}$	$O(n)$ (reject.)	$O(n^2)$ (reject.)
8.	Filaments	$\mathcal{F}$	$O(n)$ (reject.)	$O(n^{3/2})$ (reject.)
9.	Mappings	$\mathcal{M}$	$O(n)$ (point.)	$O(n^2)$ (point.; sing. ceil.)
10.	Fountains	$\mathcal{O}$	$O(n)$ (reject.)	$O(n)$ (sing. seq.)
11.	Weighted Dyck	$\mathcal{E}$	$O(n)$ (reject.)	$O(n)$ (sing. seq.)

FIGURE 10. The best strategies of the paper for Boltzmann sampling: rejection (Section 6.1, 6.2), pointing (Section 6.3), singular sequence (Section 7.1), and singular ceiled (Section 7.2).

algorithms for permutations can be adapted to obtain derangements ($\mathfrak{P}(\mathfrak{C}_{\geq 2}(\mathcal{Z}))$ and the like) and involutions ($\mathfrak{P}(\mathfrak{C}_{1,2}(\mathcal{Z}))$ and related structures); the branching process algorithms deduced automatically for binary trees apply equally well to unbalanced 2–3 trees ($\mathcal{U} = \mathcal{Z} + \mathcal{U}^2 + \mathcal{U}^3$) and to other families of trees defined by degree restrictions; random mappings satisfying various constraints then become amenable to Boltzmann sampling, and so on.

This article has shown that combinatorial samplers can be *compiled automatically* from formal specifications ("grammars") describing combinatorial models. The process is an efficient one as the program size of the sampler is derived by a single-pass linear-time formal transformation. A general-purpose implementation would most conveniently be developed on top of MAPLE's Combstruct, as many functionalities are already available there. As matter of fact, a prototype has been developed by Marni Mishna; together with other experiments, it confirms the ease of implementation and the practical efficiency of Boltzmann sampling for the random generation of many different types of combinatorial structures.

In forthcoming works, we propose to demonstrate the versatility of Boltzmann sampling for a number of simulation needs including:

— the extension of the set of allowed constructions, e.g., in the unlabelled case, sampling for multisets ($\mathfrak{M}$, repetitions are allowed), powersets ($\mathfrak{P}$, no repetitions allowed), cycles ($\mathfrak{C}$), and the substitution operation;

— multivariate extensions, meaning the sampling of configurations according to a constraint on size *and* on an auxiliary parameter (e.g., words of some length containing an unusual number of occurrences of a designated pattern);

— the realization of Boltzmann samplers using only discrete sources of randomness and basic logical operations in the style of Knuth and Yao's fundamental study [40]—nearly linear boolean (bit level) complexity still seems to be achievable in many cases of practical interest.

Acknowledgements: The authors are grateful to Alain Denise, Bernard Ycart, Brigitte Vallée, Jim Fill, Marni Mishna, Paul Zimmermann, and Philippe Robert for bibliographical suggestions, programming ideas, as well as encouragements and architectural remarks.

This work was supported in part by the ALCOM–FT Project IST-1999-14186 of the European Union. *Merci* also to Gilles Kahn and INRIA for backing the ALCOPHYS Action under which some of these ideas were hatched and to CNRS (GDR ALP and Department STIC) for its sustained support of the French Group ALÉA. *Grazie mille* finally to Alberto del Lungo and Renzo Pinzani for kindly offering an occasion to expose an early form of these ideas at the GASCOM meeting, Sienna, November 2001.

References

[1] AHO, A. V., AND CORASICK, M. J. Efficient string matching: an aid to bibliographic search. *Communications of the ACM 18* (1975), 333–340.

[2] ARNEY, J., AND BENDER, E. D. Random mappings with constraints on coalescence and number of origins. *Pacific Journal of Mathematics 103* (1982), 269–294.

[3] BARCUCCI, E., PINZANI, R., AND SPRUGNOLI, R. The random generation of directed animals. *Theoretical Computer Science 127*, 2 (1994), 333–350.

[4] BERGERON, F., LABELLE, G., AND LEROUX, P. *Combinatorial species and tree-like structures*. Cambridge University Press, Cambridge, 1998.

[5] BRENT, R. P., AND POLLARD, J. M. Factorization of the eighth Fermat number. *Mathematics of Computation 36* (1981), 627–630.

[6] BURRIS, S. N. *Number theoretic density and logical limit laws*, vol. 86 of *Mathematical Surveys and Monographs*. American Mathematical Society, Providence, RI, 2001.

[7] COMPTON, K. J. A logical approach to asymptotic combinatorics. I. First order properties. *Advances in Mathematics 65* (1987), 65–96.

[8] COMPTON, K. J. A logical approach to asymptotic combinatorics. II. Second–order properties. *Journal of Combinatorial Theory,* Series A *50* (1987), 110–131.

[9] COMTET, L. *Advanced Combinatorics*. Reidel, Dordrecht, 1974.

[10] DEMBO, A., VERSHIK, A., AND ZEITOUNI, O. Large deviations for integer partitions. *Markov Processes and Related Fields 6*, 2 (2000), 147–179.

[11] DEN HOLLANDER, F. *Large deviations*. American Mathematical Society, Providence, RI, 2000.

[12] DENISE, A., DUTOUR, I., AND ZIMMERMANN, P. CS: a MuPAD package for counting and randomly generating combinatorial structures. In *Proceedings of 10th Conference on Formal Power Series and Algebraic Combinatorics (FPSAC'98)* (1998), pp. 195–204.

[13] DENISE, A., AND ZIMMERMANN, P. Uniform random generation of decomposable structures using floating-point arithmetic. *Theoretical Computer Science 218*, 2 (1999), 233–248.

[14] DEVROYE, L. *Non-Uniform Random Variate Generation*. Springer Verlag, 1986.

[15] DOMB, C., AND BARRETT, A. Enumeration of ladder graphs. *Discrete Mathematics 9* (1974), 341–358.

[16] DRMOTA, M. Systems of functional equations. *Random Structures & Algorithms 10*, 1–2 (1997), 103–124.

[17] DUCHON, P. Relaxed random generation of trees. *Algorithms Seminar*, 05-11-01, 2001.

[18] DUCHON, P., FLAJOLET, P., LOUCHARD, G., AND SCHAEFFER, G. Random sampling from Boltzmann principles. In *Automata, Languages, and Programming* (2002), P. Widmayer et al., Ed., no. 2380 in Lecture Notes in Computer Science, Springer Verlag, pp. 501–513.

[19] DUTOUR, I., AND FÉDOU, J.-M. Object grammars and random generation. *Discrete Mathematics and Theoretical Computer Science 2* (1998), 47–61.

[20] FELLER, W. *An Introduction to Probability Theory and its Applications*, third ed., vol. 1. John Wiley, 1968.

[21] FELLER, W. *An Introduction to Probability Theory and Its Applications*, vol. 2. John Wiley, 1971.

[22] FILL, J. A., AND HUBER, M. The randomness recycler: A new technique for perfect sampling. In *Proceeding of the 41th Annual IEEE Symposium on Foundations of Computer Science* (2000), pp. 503–511.

[23] FLAJOLET, P. Combinatorial aspects of continued fractions. *Discrete Mathematics 32* (1980), 125–161.

[24] FLAJOLET, P., AND NOY, M. Analytic combinatorics of non-crossing configurations. *Discrete Mathematics 204*, 1-3 (1999), 203–229. (Selected papers in honor of Henry W. Gould).

[25] Flajolet, P., and Odlyzko, A. M. Random mapping statistics. In *Advances in Cryptology* (1990), J.-J. Quisquater and J. Vandewalle, Eds., vol. 434 of *Lecture Notes in Computer Science*, Springer Verlag, pp. 329–354. Proceedings of Eurocrypt'89, Houtalen, Belgium, April 1989.

[26] Flajolet, P., and Odlyzko, A. M. Singularity analysis of generating functions. *SIAM Journal on Algebraic and Discrete Methods 3*, 2 (1990), 216–240.

[27] Flajolet, P., Salvy, B., and Zimmermann, P. Automatic average–case analysis of algorithms. *Theoretical Computer Science 79*, 1 (Feb. 1991), 37–109.

[28] Flajolet, P., and Sedgewick, R. *Analytic Combinatorics*. 2001. Book in preparation: Individual chapters are available as INRIA Research Reports 1888, 2026, 2376, 2956, 3162, 4103 and electronically under http://algo.inria.fr/flajolet/Publications/books.html.

[29] Flajolet, P., Zimmerman, P., and Van Cutsem, B. A calculus for the random generation of labelled combinatorial structures. *Theoretical Computer Science 132*, 1-2 (1994), 1–35.

[30] Goulden, I. P., and Jackson, D. M. *Combinatorial Enumeration*. John Wiley, New York, 1983.

[31] Gourdon, X. Largest component in random combinatorial structures. *Discrete Mathematics 180*, 1-3 (1998), 185–209.

[32] Greene, D. H. *Labelled formal languages and their uses*. PhD thesis, Stanford University, June 1983. Available as Report STAN-CS-83-982.

[33] Greene, D. H., and Knuth, D. E. *Mathematics for the analysis of algorithms*. Birkhäuser, Boston, 1981.

[34] Harary, F., and Palmer, E. M. *Graphical Enumeration*. Academic Press, 1973.

[35] Hayman, W. K. A generalization of Stirling's formula. *Journal für die reine und angewandte Mathematik 196* (1956), 67–95.

[36] Howell, J. A., Smith, T. F., and Waterman, M. S. Computation of generating functions for biological molecules. *SIAM Journal on Applied Mathematics 39*, 1 (1980), 119–133.

[37] Huang, K. *Statistical Mechanics*, 2nd ed. John Wiley, 1987.

[38] Johnson, N. L., and Kotz, S. *Discrete Distributions*. John Wiley, 1969.

[39] Knuth, D. E. *The Art of Computer Programming*, 3rd ed., vol. 2: Seminumerical Algorithms. Addison-Wesley, 1998.

[40] Knuth, D. E., and Yao, A. C. The complexity of nonuniform random number generation. In *Algorithms and complexity (Proc. Sympos., Carnegie-Mellon Univ., Pittsburgh, Pa., 1976)*. Academic Press, New York, 1976, pp. 357–428.

[41] Lalley, S. P. Finite range random walk on free groups and homogeneous trees. *Ann. Probab. 21*, 4 (1993), 2087–2130.

[42] Lothaire, M. *Combinatorics on Words*, vol. 17 of *Encyclopedia of Mathematics and its Applications*. Addison–Wesley, 1983.

[43] Louchard, G. Probabilistic analysis of some (un)directed animals. *Theoretical Computer Science 159*, 1 (1996), 65–79.

[44] Louchard, G. Probabilistic analysis of column-convex and directed diagonally-convex animals. *Random Structures & Algorithms 11*, 2 (1997), 151–178.

[45] Louchard, G. Asymptotic properties of some underdiagonal walks generation algorithms. *Theoretical Computer Science 218*, 2 (1999), 249–262.

[46] Louchard, G. Probabilistic analysis of column-convex and directed diagonally-convex animals. II. Trajectories and shapes. *Random Structures & Algorithms 15*, 1 (1999), 1–23.

[47] Lyons, R., Pemantle, R., and Peres, Y. Conceptual proofs of $L \log L$ criteria for mean behavior of branching processes. *The Annals of Probability 23*, 3 (1995), 1125–1138.

[48] Martin, R., and Randall, D. Sampling adsorbing staircase walks using a new Markov chain decomposition method. In *Proc. of the 41st Annual IEEE Symposium on Foundations of Computer Science (FOCS 2000)* (2000), pp. 492–502.

[49] Meir, A., and Moon, J. W. On the altitude of nodes in random trees. *Canadian Journal of Mathematics 30* (1978), 997–1015.

[50] Milenković, O., and Compton, K. J. Probabilistic transforms for combinatorial urn models. Preprint, 2002.

[51] Nijenhuis, A., and Wilf, H. S. *Combinatorial Algorithms*, second ed. Academic Press, 1978.

[52] Odlyzko, A. M. Asymptotic enumeration methods. In *Handbook of Combinatorics*, R. Graham, M. Grötschel, and L. Lovász, Eds., vol. II. Elsevier, Amsterdam, 1995, pp. 1063–1229.

[53] ODLYZKO, A. M., AND WILF, H. S. The editor's corner: n coins in a fountain. *American Matematical Monthly 95* (1988), 840–843.
[54] OLVER, F. W. J. *Asymptotics and Special Functions*. Academic Press, 1974.
[55] QUISQUATER, J.-J., AND DELESCAILLE, J.-P. How easy is collision search? Application to DES. In *Proceedings of EUROCRYPT'89* (1990), vol. 434 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 429–434.
[56] SCHAEFFER, G. Random sampling of large planar maps and convex polyhedra. In *Proceedings of the thirty-first annual ACM symposium on theory of computing (STOC'99)* (Atlanta, Georgia, May 1999), ACM press, pp. 760–769.
[57] SHEPP, L. A., AND LLOYD, S. P. Ordered cycle lengths in a random permutation. *Transactions of the American Mathematical Society 121* (1966), 340–357.
[58] SLOANE, N. J. A. *The On-Line Encyclopedia of Integer Sequences*. 2000. Published electronically at `http://www.research.att.com/~njas/sequences/`.
[59] SORIA-COUSINEAU, M. *Méthodes d'analyse pour les constructions combinatoires et les algorithmes*. Doctorat ès sciences, Université de Paris–Sud, Orsay, July 1990.
[60] STANLEY, R. P. *Enumerative Combinatorics*, vol. I. Wadsworth & Brooks/Cole, 1986.
[61] STANLEY, R. P. *Enumerative Combinatorics*, vol. II. Cambridge University Press, 1998.
[62] STEIN, P. R., AND WATERMAN, M. S. On some new sequences generalizing the Catalan and Motzkin numbers. *Discrete Mathematics 26*, 3 (1979), 261–272.
[63] VAN CUTSEM, B. Combinatorial structures and structures for classification. *Computational Statistics & Data Analysis 23*, 1 (1996), 169–188.
[64] VAN CUTSEM, B., AND YCART, B. Indexed dendrograms on random dissimilarities. *Journal of Classification 15*, 1 (1998), 93–127.
[65] VAN DER HOEVEN, J. Relax, but don't be too lazy. Preprint, 2001. 65 pages. Available at `http://www.math.u-psud.fr/~vdhoeven/`.
[66] VAN RENSBURG, E. J. J. *The statistical mechanics of interacting walks, polygons, animals and vesicles*. Oxford University Press, Oxford, 2000.
[67] VERSHIK, A. M. Statistical mechanics of combinatorial partitions, and their limit configurations. *Funktsional'nyĭ Analiz i ego Prilozheniya 30*, 2 (1996), 19–39.
[68] WEIERMANN, A. Zero-one law characterizations of ε_0. In *Mathematics and Computer Science II: Algorithms, Trees, Combinatorics and Probabilities* (Basel, 2002), B. Chauvin, P. Flajolet, D. Gardy, and A. Mokkadem, Eds., Trends in Mathematics, Birkhäuser Verlag, pp. 527–539.
[69] WILF, H. S. *Generatingfunctionology*. Academic Press, 1990.
[70] WOODS, A. R. Coloring rules for finite trees, and probabilities of monadic second order sentences. *Random Structures Algorithms 10*, 4 (1997), 453–485.
[71] ZIMMERMANN, P. Arithmétique en précision arbitraire. Research Report 4272, Institut National de Recherche en Informatique et en Automatique, Sept. 2001. 22 pages.

P. DUCHON.: LaBRI, UNIVERSITÉ DE BORDEAUX I, 351 COURS DE LA LIBÉRATION, F-33405 TALENCE CEDEX, FRANCE; duchon@labri.fr

P. FLAJOLET: ALGORITHMS PROJECT, INRIA-ROCQUENCOURT, F-78153 LE CHESNAY, FRANCE; Philippe.Flajolet@inria.fr.

G. LOUCHARD, UNIVERSITÉ LIBRE DE BRUXELLES, DÉPARTEMENT D'INFORMATIQUE, BOULEVARD DU TRIOMPHE, B-1050 BRUXELLES, BELGIQUE; louchard@ulb.ac.be

G. SCHAEFFER: ADAGE GROUP, LORIA – CNRS, 615 RUE DU JARDIN BOTANIQUE, F-54000 VILLERS-LES-NANCY, FRANCE. Gilles.Schaeffer@loria.fr