Dirichlet Random Samplers for Multiplicative Structures

Olivier Bodini[*] Jérémie Lumbroso[†‡]

Abstract

In 2001, Duchon, Flajolet, Louchard and Schaeffer introduced Boltzmann samplers, a radically novel way to efficiently generate huge random combinatorial objects without any preprocessing; the insight was that the probabilities can be obtained directly by evaluating the generating functions of combinatorials classes. Over the following decade, a vast array of papers has increased the formal expressiveness of these random samplers.

Our paper introduces a new kind of sampler which generates *multiplicative* combinatorial structures, which enumerated by Dirichlet generating functions. Such classes, which are significantly harder to analyze than their additive counterparts, are at the intersection of combinatorics and analytic number theory. Indeed, one example we fully discuss is that of *ordered factorizations*. While we recycle many of the concepts of Boltzmann random sampling, our samplers no longer obey a Boltzmann distribution; we thus have coined a new name for them: *Dirichlet samplers*. These are very efficient as they can generate objects of size n in $O((\log n)^2)$ worst-case time complexity.

By providing a means by which to generate very large random multiplicative objects, our Dirichlet samplers can facilitate the investigation of these interesting, yet little studied structures. We also hope to illustrate some of our general ideas regarding the future direction for random sampling.

Introduction

Nijenhuis and Wilf introduced the recursive method [24] in the late 70s (later extended by [12]), the first automatic random generation method; it is termed *automatic* because it can directly derive random samplers from any combinatorial description—no bijection, no clever algorithm, no complicated equations are needed. The drawback is that this method is costly: to generate an object of size n, it requires knowing the complete enumeration of the combinatorial class up to size n; and predictably when n is large, this enumeration is significant both to calculate and to store.

Enter Boltzmann sampling, introduced by Duchon *et al.* in 2002 [5, 6], of which the key insight was that the coefficients do not need to be extracted: instead, correct probabilities can be obtained by proxy, by evaluating the generating functions in $O(1)$[1].

As a consequence, through evaluation, all the coefficients of a generating function are smashed together, and the resulting probabilities take into account objects of *all sizes*. Thus, though you *do* know that the object returned will be *uniformly* sampled among objects of the same size, the size itself is a random variable—which you have no direct control over. As a result, a significant aspect of Boltzmann sampling involves: (a) rejecting objects which are not within the desired size interval; (b) manipulating the generating functions so the size distribution is such that not too many objects need be rejected.

The efficiency of this approach, combined with its mathematical appeal, have made it a fertile topic, and many of its aspects have been developed: from the expressiveness of the specifications that can be handled [8, 14, 1], to the way generating functions are evaluated [25, 26], and the way discrete probabilities are sampled from [10].

Multiplicative combinatorics. In this paper, we extend the tenets of Boltzmann sampling to multiplicative combinatorial classes. And while the main premise—of using the generating functions as shortcuts to calculate probabilities—remains the same, multiplicative combinatorial classes bring with them a great deal of new challenges.

Perhaps most significantly, the size distributions are wildly different, and the expected value of the size of the objects which are drawn is *infinite*. This completely changes how we approach the choice of the control parameter (the value at which generating functions are evaluated), makes exact generation a pipe dream, and makes anticipated rejection crucial.

Outline of this paper. In Section 1, we give the obligatory definitions and constructions which are the basis of our Dirichlet samplers; to those familiar with Boltzmann samplers, much of this will feel trite, but

[*]LIPN, UMR 7030, Un. Paris 13/CNRS, F-93430 Villetaneuse.
[†]Algorithms, INRIA Rocq., 78153 Le Chesnay, France.
[‡]APR, LIP6/UPMC, 4 place Jussieu, 75005 Paris, France.

[1]Initially, constant time evaluation of the generating functions was only conjectured, but Pivoteau *et al.* [25, 26] confirmed it is possible given the combinatorial origin of these functions.

Subsection 1.4 already highlights a few of differences. Section 2 goes into detail on the polylogarithmic worst-case time complexity of Dirichlet random generation; in particular, we give two central theorems on the rejection complexity given the particular size distributions of multiplicative combinatorial classes, and do so appealing to [3], a classical theorem of analytic number theory. We then present extensive examples in Section 3. This allows us to cover broad aspects of our Dirichlet samplers, partially situate our research within the existing number theoretical context, and provide a glimpse of multiplicative combinatorial objects.

Finally, the close interactions between additive combinatorial classes, their generating functions and Boltzmann samplers has persistently covered up the fact that these are three different types of objects—and that while they are related, they are not bound to each other. This is a concept that is all the more crucial in multiplicative combinatorial classes, because the ties are not so strong anymore; in this article, this is mostly developed in Section 4, in which we introduce an algebra for our samplers. Section 5 wraps everything up and is the opportunity to sketch broad future developments.

1 Main Concepts

This section presents the main definitions and concepts investigated in this paper, and it is divided in four parts. First, we give a succinct overview of multiplicative combinatorics (for a more detailed account, see [16, §8]) and how Dirichlet generating functions naturally encode multiplicative objects. We then introduce our main object of study, Dirichlet samplers, and precisely state their distributional properties, after which we detail the base constructions of these samplers. Finally, we explicitly mention the challenges which arise, and discuss where Dirichlet samplers are distinct from their Boltzmann counterparts.

Throughout this section, a few notions of additive combinatorics might be helpful, and we assume our reader is familiar with the *symbolic method*—a simple set of rules for formally specifying combinatorial classes and automatically translating this specification to counting generating functions. The *de facto* reference on this topic is Flajolet and Sedgewick's book, and should our reader need it, we highly recommend the short introduction given by [11, I].

1.1 Multiplicative Combinatorics. We consider *multiplicative* combinatorial classes, endowed with a size function $|\cdot|$ which assigns every object of a class a value in $\mathbb{Z}_{>0}$ (note the strict positivity)—with the condition that, while a class may contain infinitely many objects, it may only contain a *finite* number of objects of any given size. These classes are called multiplicative because the size of the Cartesian product of two of their objects is the *product* of the size of the two composite objects (and not the *sum* as is traditionally the case in additive combinatorial structures).

Dirichlet Generating Functions. For instance, consider a class $\mathcal{A}$ defined as the Cartesian product of two other classes, that is $\mathcal{A} = \mathcal{B} \times \mathcal{C}$[2]. If $\beta \in \mathcal{B}$ and $\gamma \in \mathcal{C}$, then the pair $\alpha = (\beta, \gamma) \in \mathcal{A}$, and $|\alpha| = |\beta| \cdot |\gamma|$. Furthermore, if we respectively denote by a_n, b_n and c_n the number of elements of size n in $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$, these are related by

$$a_n = \sum_{d|n} b_d \cdot c_{n/d} \tag{1.1}$$

which is a *Dirichlet convolution*. Thus multiplicative combinatorial classes are naturally enumerated by Dirichlet generating functions.

Definition 1. *Let $\mathcal{A}$ be a multiplicative combinatorial class, and a_n the number of objects from $\mathcal{A}$ that have size n. The* Dirichlet generating function *(DGF) associated with class $\mathcal{A}$ is defined equivalently by*

$$A(s) = \sum_{n=0}^{\infty} a_n n^{-s} \qquad \textit{or} \qquad A(s) = \sum_{\alpha \in \mathcal{A}} |\alpha|^{-s}.$$

We often refer to the largest real singularity *of $A(s)$, which we always note ρ (as is customarily the case).*

An important consequence of this multiplicativity is that the class $\mathcal{Z}$ (containing only a single object of unit size) which held such an important role of *atomic class* in additive combinatorics, plays the part of *neutral element* in multiplicative combinatorics—while we now deal with *infinitely many atoms* as, indeed, there is a different atom $\mathcal{Z}_p$ for every prime p.

The main constructions for multiplicative combinatorial classes, as well as their translation to Dirichlet generating functions, are listed in Table 1. Our work on Dirichlet random samplers has already been extended to Pólya operators (cycle, multiset, etc.) necessary, for instance, to define *unordered* factorizations on primes; but we have chosen, for length considerations, to postpone that topic to an extended version of this paper.

[2]This notation means the class $\mathcal{A}$ is the Cartesian product of classes $\mathcal{B}$ and $\mathcal{C}$, meaning it is composed of all possible ordered pairs in which the first element is an object taken from $\mathcal{B}$ and the second element is an object taken from $\mathcal{C}$; should this baffle you, refer to [11, I.2].

Class	*Description*	*Dirichlet Generating Function*
$\varepsilon = \mathcal{Z}_1$	Neutral element	1
$\mathcal{Z}_p \quad (p \in \mathbb{P})$	Atomic elements	p^{-s}
$\mathcal{I} = \bigcup_{k=1}^{\infty} \mathcal{Z}_k$	Integer class	$I(s) = \zeta(s)$
$\mathcal{P} = \bigcup_{p\in\mathbb{P}} \mathcal{Z}_p$	Prime class[3]	$P(s) = \sum_{p\in\mathbb{P}} p^{-s}$
$\mathcal{A} = \mathcal{B} + \mathcal{C}$	Disjoint union	$A(s) = B(s) + C(s)$
$\mathcal{A} = \mathcal{B} \times \mathcal{C}$	Cartesian product	$A(s) = B(s) \cdot C(s)$
$\mathcal{A} = \textsc{Seq}\,(\mathcal{B})$	Sequence	$A(s) = 1/(1 - B(s))$

Table 1. Elementary multiplicative combinatorial constructions and their translation to Dirichlet generating functions; we have here called $\mathbb{P}$ the set of primes (this notation is not used anywhere else in this paper).

1.2 Dirichlet Random Samplers. Having introduced the main concepts of multiplicative combinatorics and Dirichlet generating functions, we now define the topic of this paper.

Definition 2. *A* Dirichlet sampler *for a multiplicative combinatorial class $\mathcal{A}$ is an algorithm which samples an object $\alpha \in \mathcal{A}$ with probability*

$$\mathbb{P}_s[\alpha] = \frac{1}{A(s)} \cdot |\alpha|^{-s}$$

where the normalizing factor $A(s)$ is the Dirichlet generating function associated with class $\mathcal{A}$, evaluated at some s called the control parameter*. Moreover we denote by $\Gamma\mathrm{D}_s[\mathcal{A}]$ the Dirichlet sampler associated with class $\mathcal{A}$.*

As is the case with Boltzmann samplers, the size of objects drawn with a Dirichlet sampler is a random variable (which we usually denote N), however the constructed objects are uniformly sampled from among all other objects of the same size. This is made explicit by the following proposition.

Proposition 1.1. *Let $\mathcal{A}$ be a multiplicative combinatorial class. If a Dirichlet sampler for $\mathcal{A}$ returns an object α with size $n = |\alpha|$, then α was drawn uniformly among all other objects of size n; in other words, the probability of drawing an object conditioned on its size is*

$$\mathbb{P}_s[\alpha \mid |\alpha| = n] = \frac{1}{a_n}$$

where, as previously, $a_n = [n^{-s}]A(s) = card(\mathcal{A}_n)$ is the number of objects of $\mathcal{A}$ which have size n.

Proof. Follows from the definition of Dirichlet samplers and conditional probabilities. First, consider that the probability of drawing an object of size n under the Dirichlet model is

$$\mathbb{P}_s[N = n] = \sum_{\alpha\in\mathcal{A}_n} \mathbb{P}_s[\alpha] = \frac{a_n n^{-s}}{A(s)}. \tag{1.2}$$

Second, consider that the probability of drawing a *specific* object of size n (from among all objects) is

$$\mathbb{P}_s[\alpha \cap |\alpha| = n] = \frac{n^{-s}}{A(s)}. \tag{1.3}$$

Then simply combining (1.2) and (1.3) following the standard definition of conditional probabilities yields

$$\mathbb{P}_s[\alpha \mid |\alpha| = n] = \frac{\mathbb{P}_s[\alpha \cap |\alpha| = n]}{\mathbb{P}_s[N = n]} = \frac{n^{-s}}{A(s)} \cdot \frac{A(s)}{a_n n^{-s}} = \frac{1}{a_n}.$$

□

1.3 Base Constructions of Dirichlet Samplers. The basic multiplicative classes and constructions, and their translation to Dirichlet, are listed in Table 1. The proofs of these equivalences are straightforward.

We will just stress that the sequence construction assumes the base class does not contain unit size objects, that is for $\mathcal{A} = \textsc{Seq}\,(\mathcal{B})$ to be a properly defined class, we must have $b_1 = 0$ (recall that b_0 is the number of elements of $\mathcal{B}$ with size 1). The combinatorial explanation for this is that if an atom which does not contribute to the size—in additive combinatorics, this is the atom of size 0; in multiplicative combinatorics, this is the atom of size 1—then there is no relation between the size and the number of atoms; as a consequence, there are sizes for which there is infinitely many different objects, which is not allowed.

To construct Dirichlet samplers, we will need to be able to draw random variates from the Bernoulli and

[3] The Dirichlet generating function for the Prime class seems to have been first studied by Glaisher in 1891, and was named *Prime Zeta function* and noted $P(s)$ by Fröberg in [13].

geometric distributions[4], respectively denoted $\mathrm{Ber}(p)$ and $\mathrm{Geo}(p)$, are defined, for $p \in (0,1)$, by

$$B \in \mathrm{Ber}(p) \qquad \mathbb{P}[B=0] = (1-p) \qquad \mathbb{P}[B=1] = p$$

and, for $k \in \mathbb{Z}_{\geqslant 0}$,

$$G \in \mathrm{Geo}(p) \qquad \mathbb{P}[G=k] = (1-p)p^k.$$

There are many ways to implement these distributions: see [6, §5] for the traditional way this has been done in Boltzmann samplers; see [4] for a more complete discussion on efficient techniques (for instance truncating an exponential variate as a means to obtain a geometric variate); or see the ongoing [10] for how to generate these random variates by purely discrete means, without doing any arithmetic.

Now, let $\mathcal{B}$ and $\mathcal{C}$ be two combinatorial classes for which (possibly recursive) Dirichlet samplers are known, and respectively noted $\Gamma\mathrm{D}_s[\mathcal{B}]$ and $\Gamma\mathrm{D}_s[\mathcal{C}]$.

Integer and prime classes. Efficient algorithms to draw atoms from these classes are given in Subsection 2.1. Infinite subsets of these classes can often satisfyingly enough be implemented by rejection. For instance, to generate the ordered factorizations discussed in Subsection 3.1, we require sampling from the class $\mathcal{I} \setminus \mathcal{Z}_1$ (all integers except 1); this can be done by sampling from $\mathcal{I}$ (using ZetaLaw) until the returned element is distinct from 1.

Finite sets. If $\mathcal{A}$ is finite, it is straightforward to select its elements according to the Dirichlet distribution explicitly provided by Definition 2 (in the previous page). In practice, $\mathcal{A}$ is usually reduced to a single element, which can be deterministically returned.

Disjoint union. If $\mathcal{A} = \mathcal{B} + \mathcal{C}$ is the union of disjoint copies of $\mathcal{B}$ and $\mathcal{C}$, its Dirichlet sampler is a mixture of the models associated to $\mathcal{B}$ and $\mathcal{C}$ steered by a Bernoulli variate:

$\Gamma\mathrm{D}_s[\mathcal{A}] := \{$
  $p_{\mathcal{A}} \leftarrow B(s)/A(s)$
  if $\mathrm{Ber}(p_{\mathcal{A}}) = 1$ **then** return $\Gamma\mathrm{D}_s[\mathcal{B}]$
  else return $\Gamma\mathrm{D}_s[\mathcal{C}]$
$\}$

Cartesian product. If $\mathcal{A} = \mathcal{B} \times \mathcal{C}$ is the Cartesian product of $\mathcal{B}$ and $\mathcal{C}$, its Dirichlet sampler returns a pair of *independently* drawn objects from $\mathcal{B}$ and $\mathcal{C}$:

$\Gamma\mathrm{D}_s[\mathcal{A}] := \{$
  return $(\Gamma\mathrm{D}_s[\mathcal{B}], \Gamma\mathrm{D}_s[\mathcal{C}])$
$\}$

[4]Note that our definition for the geometric distribution, while the same used by Boltzmann samplers, is non standard (and, for instance, when using a computational software program, you will have to generate geometric variates of parameter $1-p$).

Sequence. If $\mathcal{A} = \mathrm{SEQ}(\mathcal{B})$ is composed of all finite sequences of elements of $\mathcal{B}$, its Dirichlet sampler draws a geometric variable K and returns a list containing K *independently* drawn copies of $\mathcal{B}$:

$\Gamma\mathrm{D}_s[\mathcal{A}] := \{$
  $K \in \mathrm{Geo}(A(x))$;
  return $(\underbrace{\Gamma\mathrm{D}_s[\mathcal{B}], \ldots, \Gamma\mathrm{D}_s[\mathcal{B}]}_{K \text{ times}})$
$\}$

As has been pointed out from the beginning [5], since the recursive calls for the Cartesian product and sequence constructions are independent, they may thus be executed concurrently. This has the potential to mightily speed up calculations on larger computing clusters.

Recursive classes. All the previously given constructions are still valid when dealing with a recursively specified class—a typical example of such a class, branching factorizations, is given in Subsection 3.3—with the only difference that the Dirichlet samplers are themselves recursive; there are few constraints for what represents a 'well-founded' recursive definitions, but these are standard (for instance, the recursion must build larger objects from strictly smaller ones).

1.4 A Word on the Size Distribution. As with Boltzmann samplers, we can calculate the expected value of the size. Let $\mathcal{A}$ be a multiplicative class, and $A(s)$ its Dirichlet generating function, the expected value of the size N of an object sampled from $\mathcal{A}$ with a Dirichlet sampler is

$$\mathbb{E}_s[N] = \frac{\sum_{\gamma \in \mathcal{A}} |\alpha|^{-s} \cdot |\alpha|}{A(s)} = \frac{A(s-1)}{A(s)}.$$

The usefulness of this result, however, is debatable. Indeed, because s is often chosen between ρ—the largest positive singularity of the DGF—and $\rho+1$, the expected value is infinite[5]. This phenomena, illustrated in Figure 1, is a side-effect of the slowly decreasing nature of the probability distribution (indeed, the n^{-s} are polynomial, whereas the ordinary and exponential generating functions of Boltzmann samplers deal with x^n which are exponential, yielding exponential tails). It has two concrete consequences: first, *exact-size* sampling is impossible—or cannot be achieved with the same technique used in conjunction with Boltzmann sampling; second, *anticipated rejection* is absolutely crucial.

No Exact-Size? As mentioned in the introduction, Boltzmann and Dirichlet sampling both make use of the following principle. To sample an object with size

[5]Although note that Dirichlet samplers can be shown to halt with probability 1.

within $[(1-\varepsilon)n, (1+\varepsilon)n]$, for some $\varepsilon \in (0,1)$, an object is first drawn with a random size (some measures may be taken to calibrate this random size as best as possible): if the object's size is within the desired interval then the sampling has succeeded; if not it is *rejected* and a new object is sampled—which will then, in turn, possibly also be rejected, etc.. In the following section, we analyze the precise cost of this rejection, which is the number of objects which are rejected before a suitable one is constructed, for Dirichlet sampling.

Exact-size sampling, in the Boltzmann model, is straightforward: it consists in rejecting objects until they are of the exact size—at a cost of rejecting a lot of objects (but still this complexity may be no worse than previous costly exact-size sampling techniques). In the Dirichlet model, the size varies so wildly that the cost of such a method is prohibitive: thus exact-size sampling is either not possible through Dirichlet sampling, or will have to be achieved through other means (Section 4 tentatively introduces a tool which might be the first step towards solving this problem).

Anticipated Rejection. In the context of random sampling, anticipated rejection is a concept which consists in keeping track, *during* the construction of an object, of its size; and if ever the size goes beyond the targeted interval, immediately stop the construction of the object and start anew (since it would eventually end up being discarded anyway). As stated in the caption of Figure 1, when randomly sampling branching factorizations of size close to 10^3, some objects can be huge, up to some 10^{242}—without anticipated rejection, these objects make random sampling in our model utterly inefficient. By contrast, with anticipated rejection, we can guarantee that, when targeting an object of size n, all objects (even those which are discarded) will be constructed in time/space, linear to the number of atoms, allowing us to give our final complexity bounds.

2 Complexity Analysis

First, we state a general theorem, which is a consequence of the analogous theorem, [6, Theorem 3.1], for Boltzmann sampling.

THEOREM 2.1. *Sampling an object with a Dirichlet sampler (without rejection) is linear in the number of atoms in the output.*

Next, the complexity analysis consists in first showing that it is reasonably efficient to sample from the Zeta distribution (which is at the heart of Dirichlet samplers), then in bounding the average number of rejected objects. Ultimately, we combine these elements to prove that an object of size n can be generated in worst-case $\Theta((\log n)^2)$ time complexity.

2.1 Sampling from the Zeta Distribution. The integers of the atomic class $\mathcal{I}$ are central to Dirichlet random samplers, and are distributed according to the *Zeta distribution.*

DEFINITION 3. *The* Zeta distribution *(sometimes also referred to as the* Zipf distribution*, of which it is a*

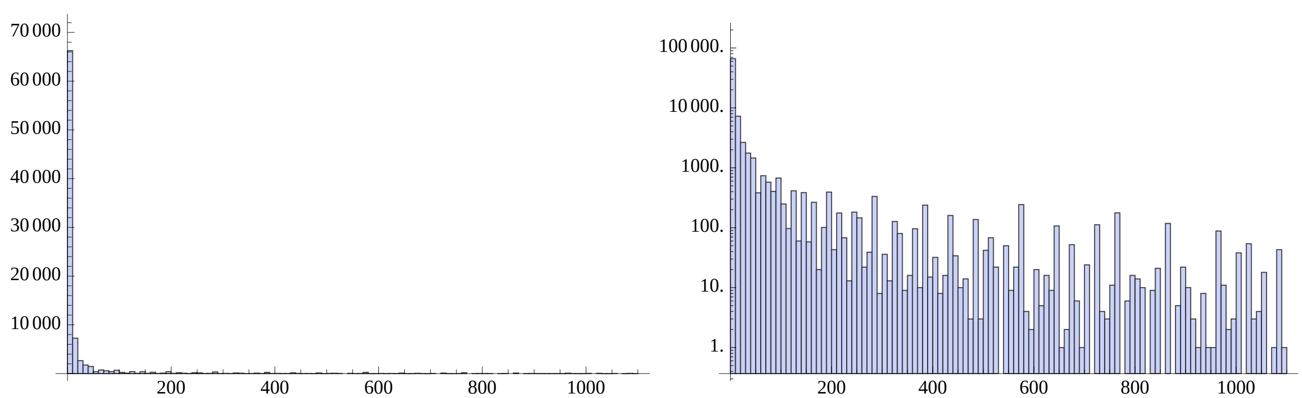

Figure 1. We generated 100 000 branching factorizations without rejection, but targeting a size of 10^3 and looked at the distribution of the sizes; it is very chaotic: the mean is around $1.48 \cdot 10^{137}$, the maximum size around $1.48 \cdot 10^{142}$ while the median is 4. Visualizing the distribution properly requires trimming the data set. Thus by restricting ourselves to only those 87 810 which have a size smaller or equal to 1100, we get the histogram on the left (x-axis is the size, and y-axis the number of elements generated of a given size); if we take a logarithmic scale for the y-axis, we notice that the distribution is practically never empty—though there are proportionally few objects of sizes 50 through 1000.

special case) is a discrete probability distribution that has one real-valued parameter $s \in (1, +\infty)$. Let X be zeta-distributed with parameter s, then

$$\mathbb{P}[X = k] := \frac{k^{-s}}{\zeta(s)}$$

where $\zeta(s) := \sum_{k=0}^{\infty} k^{-s}$ is the Riemann zeta function.

Of course, efficiently drawing an integer following this probability distribution is paramount to Dirichlet random generation. On this topic, we refer to a sampler using the rejection method[6], and suggested by Devroye in [4], his seminal book on non-uniform random variate generation.

```
function ZetaLaw(s)
    b ← 2^(s−1)
    repeat
        U ← random(0, 1) and V ← random(0, 1)
        X ← ⌊U^(−1/(s−1))⌋
        T ← (1 + 1/X)^(s−1)
    until VX(T − 1)/(b − 1) ⩽ T/b
    return X
end function
```

Figure 2. Efficient rejection algorithm to randomly sample from the Zeta distribution.

Proposition 2.1. (Devroye) *The number of iterations made by the algorithm* ZetaLaw(s) *is a random variable R which has expected value*

$$\mathbb{E}_s[R] = \frac{2^{s-1}}{\zeta(s)\left(2^{s-1} - 1\right)}.$$

Therefore in particular, as s tends towards the singularity $s = 1$,

$$\lim_{s \to 1} \mathbb{E}_s[R] = \frac{1}{\log 2} \approx 1.4426....$$

A similar algorithm can be devised for the generation of the *Prime Zeta distribution*, which is the analog of the Zeta distribution where the support is prime integers instead of all strictly positive integers[7]: generate a zeta-distributed integer with ZetaLaw(s) until the returned integer is a prime.

Remark. While the algorithm referenced here for the generation of the Zeta distribution is very efficient, it resorts to arbitrary-precision arithmetic. Following the work begun in [10], it is a worthwhile research problem to determine whether the Zeta distribution can be sampled from, *solely using random bits and elementary arithmetic on integers*. Although this distribution seems out of reach of most existing methods, which rely on Taylor series (whether implicitly or explicitly), some elements, most notably the Buffon integrator, seem to indicate it may be an attainable goal.

2.2 Average Number of Discarded Objects. In this subsection, we prove that on average, we need to draw a logarithmic number of objects before obtaining one within the targeted size interval. We state two theorems: a general one that applies to a broad range of combinatorial classes; and one specifically for combinatorial classes with a square-root type singularity. This second theorem yields a slightly less good complexity bound, but we will see in Section 4 how to modify the generating functions of these classes so that they fit the general case.

We first introduce a definition, which basically regroups the conditions ensuring we may apply the Delange theorem to a given Dirichlet generating function.

Definition 4. *We say of a Dirichlet generating function $f(s)$ that it is* regularly singular *if and only if:*

- *all coefficients of $f(s)$ are positive reals;*
- *$f(s)$ is convergent for all $s > \rho > 0$ (ρ, the* largest *positive singularity of $f(s)$, is the function's* convergence point*);*
- *$f(s)$ is analytic in all points of the vertical line $x = \rho$ other than $(\rho, 0)$;*
- *in the neighborhood of $(\rho, 0)$, for $s > \rho$, we have*

$$f(s) = g(s) + \frac{h(s)}{(s - \rho)^{\alpha}} \qquad (2.4)$$

where $\alpha \in \mathbb{R} \backslash \{0, -1, -2, \ldots\}$ is called the function's singular exponent[8]*, and $g(s)$ and $h(s)$ are functions which are analytic for $s = \rho$, with $h(\rho) \neq 0$.*

Remark. It would be interesting to determine whether all Dirichlet generating functions coming from a well-founded combinatorial specification automatically verify the previous definition; or even to reduce hypotheses

[6]For those not in the know, the rejection method more or less consists in generating a distribution $f(x)$ by: finding a *dominating distribution* $g(x)$, such that $f(x) < c \cdot g(x)$, which is easy—or easier—to generate; then generating $g(x)$-distributed variates and rejecting them until one is found that happens to be $f(x)$-distributed. The efficiency of this method depends on how tightly $g(x)$ bounds $f(x)$.

[7]And the distribution is normalized by the Prime Zeta function, instead of the Riemann zeta function.

[8]As usual, when ρ is a pole, α is simply its multiplicity.

(for instance the positivity of coefficients is always trivially verified by DGF with a combinatorial origin).

THEOREM 2.2. *Let $f(s)$ be the Dirichlet generating function associated to a multiplicative combinatorial class. If $f(s)$ is regularly singular and has a strictly positive singular exponent, $\alpha > 0$, then the average rejection complexity is $\Theta(\log n)$. This is achieved by taking as control parameter*

$$s \sim \rho + \frac{\alpha}{\log n}.$$

THEOREM 2.3. *Let $f(s)$ be the Dirichlet generating function associated to a multiplicative combinatorial class. If $f(s)$ is regularly singular and has a singular exponent $\alpha = -1/2$, that is if near ρ*

$$f(s) = g(s) - h(s)\sqrt{s-\rho},$$

then the average rejection complexity is $\Theta((\log n)^{3/2})$. This is achieved by taking as control parameter

$$s \sim \rho + \frac{h(\rho)^2}{4g(\rho)^2(\log n)^2}.$$

Proof to Theorem 2.2. To bound the asymptotic number of trials (the number of objects which are rejected by the Dirichlet sampler), we must bound the probability of an object being drawn with its size within the desired interval.

Let N be the random variable of the size of a drawn object (which has not been filtered by rejection). Let $f(s)$ be the Dirichlet generating function associated with the multiplicative combinatorial class. By definition of the Dirichlet distribution,

$$\mathbb{P}_s[N = k] = \frac{a_k k^{-s}}{f(s)},$$

thus by simple summation,

$$\mathbb{P}_s\left[\left|\frac{N}{n} - 1\right| \leqslant \varepsilon\right] = \sum_{k=1}^{(1+\varepsilon)n} \frac{a_k \cdot k^{-s}}{f(s)} - \sum_{k=1}^{(1-\varepsilon)n} \frac{a_k \cdot k^{-s}}{f(s)}. \tag{2.5}$$

Under our theorem's hypotheses for $f(s)$, Delange's Tauberian theorem—from [3], but often referred to as in Tenenbaum's book [27, II.7]—yields the following asymptotic estimate (for any large M),

$$\sum_{k=1}^{M} [k^{-s}]f(s) \sim \frac{h(\rho)}{\rho \cdot \Gamma(\alpha)} M^{\rho} (\log M)^{\alpha-1}. \tag{2.6}$$

Applying the equivalent (2.6) to the sums of (2.5), we obtain

$$\sum_{k=1}^{(1\pm\varepsilon)n} \frac{a_k}{f(s)} \sim \frac{h(\rho)}{\rho \cdot \Gamma(\alpha) f(s)} ((1 \pm \varepsilon)n)^{\rho} (\log((1 \pm \varepsilon)n))^{\alpha-1}. \tag{2.7}$$

That is,

$$\sum_{k=(1-\varepsilon)n}^{(1+\varepsilon)n} \frac{a_k}{f(s)} \sim \frac{h(\rho)}{\rho \cdot \Gamma(\alpha) f(s)} \left([(1+\varepsilon)n]^{\rho} [\log((1+\varepsilon)n)]^{\alpha-1} \right.$$
$$\left. - [(1-\varepsilon)n]^{\rho} [\log((1-\varepsilon)n)]^{\alpha-1}\right).$$

By lower-bounding the k^{-s} factor by $((1+\varepsilon)n)^{-s}$, we can provide a preliminary lower bound for (2.5)

$$((1+\varepsilon)n)^{-s} \sum_{k=(1-\varepsilon)n}^{(1+\varepsilon)n} \frac{a_k}{f(s)} \leqslant \sum_{k=(1-\varepsilon)n}^{(1+\varepsilon)n} \frac{a_k \cdot k^{-s}}{f(s)}. \tag{2.8}$$

Because we can simplify the summation in (2.8) by using the equivalents of (2.7), finding an interesting lower bound now only requires to find the s that maximizes

$$\Psi(s) = ((1+\varepsilon)n)^{-s} \frac{h(\rho)}{\rho \cdot f(s)\Gamma(\alpha)} \cdot$$
$$\left[((1+\varepsilon)n)^{\rho} \cdot (\log((1+\varepsilon)n))^{\alpha-1} \right.$$
$$\left. - ((1-\varepsilon)n)^{\rho} (\log((1-\varepsilon)n))^{\alpha-1}\right].$$

Calculating the derivative of Ψ, we find such an s to be

$$s = \frac{\rho \cdot \log((1+\varepsilon)n) + \alpha}{\log((1+\varepsilon)n)} \sim \rho + \frac{\alpha}{\log n}.$$

Finally, $\Psi(s)$ needs to be approximated in $s = \rho + \alpha/\log n$, when n tends to infinity:

$$\Psi\left(\rho + \frac{\alpha}{\log n}\right) \sim \frac{\alpha^{\alpha} \cdot h(\rho)}{\rho \cdot \Gamma(\alpha) \cdot \mathrm{e}^{\alpha}} \left(1 - \left(\frac{1-\varepsilon}{1+\varepsilon}\right)^{\rho}\right) \frac{1}{\log n}. \tag{2.9}$$

We have thus obtained a lower bound for our probability. With the exact same arguments, we may obtain an analogous upper bound,

$$\frac{\alpha^{\alpha} \cdot h(\rho)}{\rho \cdot \Gamma(\alpha) \cdot \mathrm{e}^{\alpha}} \left(\left(\frac{1+\varepsilon}{1-\varepsilon}\right)^{\rho} - 1\right) \frac{1}{\log n}.$$

The end-result is the following *asymptotic* double inequality,

$$K(\alpha,\rho) \cdot \left(1 - \left(\frac{1-\varepsilon}{1+\varepsilon}\right)^{\rho}\right) \frac{1}{\log n} \geqslant$$
$$\mathbb{P}_s\left[\frac{|N-n|}{n} < \varepsilon\right] \geqslant$$
$$K(\alpha,\rho) \cdot \left(\left(\frac{1+\varepsilon}{1-\varepsilon}\right)^{\rho} - 1\right) \frac{1}{\log n}$$

with

$$K(\alpha,\rho) := \frac{\alpha^{\alpha} \cdot h(\rho)}{\rho \cdot \Gamma(\alpha) \cdot \mathrm{e}^{\alpha}}.$$

Specifically, this means that an order $\log n$ of objects must be rejected to draw an object within the desired size interval. $\square$

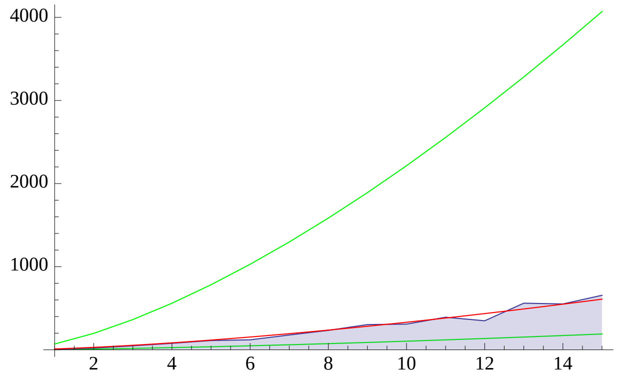

Figure 3. In blue, the evolution of the average number of discarded objects when sampling 100 branching factorizations of sizes 10^i, where i is the x-axis and goes from 1 to 15. In green, the theoretical bounds calculated for the theorem; in red, an empirical bound, $3\cdot\log(n)^{3/2}$.

Proof to Theorem 2.3. The proof is similar to the one for Theorem 2.2. The difference lies in the fact that since the singular exponent α equals $-1/2$, the term $f(s)$ in the denominator of $\Psi(s)$, as defined by equation (2.9), no longer tends to infinity when s tends to the singularity ρ, but instead converges to a constant value.

So in effect, we have $f(s) \sim g(\rho) - h(\rho)\sqrt{s-\rho}$ which means that $\Psi(s)$ is maximized when

$$s = \rho + \frac{h(\rho)^2}{4g(\rho)^2(\log n)^2} + O\left(\frac{1}{(\log n)^3}\right)$$

and

$$\Psi\left(\rho + \frac{h(\rho)^2}{4g(\rho)^2(\log n)^2}\right) \sim \frac{h(\rho)}{2\rho\sqrt{\pi}\cdot g(\rho)}\left(1 - \left(\frac{1-\varepsilon}{1+\varepsilon}\right)^{\rho}\right)\frac{1}{(\log n)^{3/2}},$$

which is a lower bound; and as before we can also obtain an upper bound, therefore concluding the proof. □

Remark. Both proofs give an upper bound on the number of trials required to randomly generate an object of size within a given interval. While this upper bound is sufficient to prove the desired order of magnitude, it is *very* large, and should not be expected to give any sort of idea on the actual average number of trials. This is illustrated in Figure 3.

2.3 Final Complexity of Constrained Generation. In this subsection, we finally piece together all elements to obtain our advertised final worst-case time complexity bound of $\Theta((\log n)^2)$, for sampling an object of size within $[(1-\varepsilon)n, (1+\varepsilon)n]$, with $\varepsilon \in (0,1)$.

THEOREM 2.4. *The Dirichlet sampler for a multiplicative combinatorial class $\mathcal{A}$, built using constructions from Table 1 and recursive definitions, samples objects of size n with a given tolerance $\varepsilon \in (0,1)$ in worst-case $O((\log n)^2)$ time complexity.*

Sketch of proof. Theorem 2.1 states that the complexity of generating an object is linear in its number of atoms. As we explained in Subsection 1.3, the sequence construction does not allow unit sized objects (as is the case, by the way, of Pólya operators for cycle, set, and other constructions) and similarly, the 'well-founded' constraint (a concept which is detailed in [6, 11]) on recursive definitions guarantees that the number of unit-size atoms in a recursion, is strongly bound. As an end result, it is possible to assume that all atoms are of size equal or larger to 2; thus the number of atoms is logarithmic in the size. Furthermore, Proposition 2.1 gives means to generate an atom from infinite atom classes in $O(1)$ time.

Combining the two previous facts, we get that the time complexity of constructing a single object (without rejection) of size n is $O(\log n)$.

Now, Theorem 2.2 states that the average number of rejections to generate an object of size within $[(1-\varepsilon)n, (1+\varepsilon)n]$, for some $\varepsilon \in (0,1)$, is $O(\log n)$ for a certain type of combinatorial classes—specifically those for which the singular exponent is strictly positive. Section 4 sketches means by which to always transform a combinatorial class into an equivalent one with such a singular exponent. Thus the rejection complexity for all classes which are considered is $O(\log n)$.

Finally, with anticipated rejection, one guarantees that all objects which are rejected are of size at most $n(1+\varepsilon)$.

To summarize, the number of rejections is $O(\log n)$, all objects are of size at most $n(1+\varepsilon)$ and the complexity of generating an object of size n is $O(\log n)$, thus the final worst-case complexity for generating an object of size n with rejection is $O((\log n)^2)$ as claimed. □

3 Dirichlet Sampling Illustrated by Examples

The present section presents three examples through which we attempt to highlight different aspects of this paper, and different *multiplicative* combinatorial structures which hopefully will not seem like the bland analogs to some additive counterparts.

3.1 Ordered Factorizations. While prime factorization has long been an important matter in cryptography, non-prime factorization has also received a lot of interest both from a number theoretic and combinatorial point of views—as a multiplicative analog to the much studied additive *integer partitions* (the summands are non-ordered) and *integer compositions* (the summands are ordered).

Thus *ordered factorizations* are the representations of a natural integer as a *product* of factors greater than one, where all permutations of the factors are considered different. As an example, there are eight

$$
\begin{array}{|l|}
\hline
\Gamma\mathrm{D}_s\,[\mathcal{F}] := \{ \\
\qquad \lambda \leftarrow \zeta(s) - 1; \\
\qquad K \in \mathrm{Geo}(\lambda); \\
\qquad \text{return } \underbrace{(\Gamma\mathrm{D}_s\,[\mathcal{I} \setminus \mathcal{Z}_1]\,, \ldots, \Gamma\mathrm{D}_s\,[\mathcal{I} \setminus \mathcal{Z}_1])}_{K \text{ times}} \\
\} \\
\hline
\end{array}
$$

Figure 4. Pseudo algorithm of a Dirichlet sampler for ordered factorizations.

ordered factorizations for 12,

$$12, \qquad 2 \cdot 6, \qquad 3 \cdot 4, \qquad 4 \cdot 3, \qquad 6 \cdot 2,$$
$$2 \cdot 2 \cdot 3, \qquad 2 \cdot 3 \cdot 2, \qquad 3 \cdot 2 \cdot 2.$$

The enumeration of ordered factorizations was first discussed by Kalmár in the early 1930s as the "*factorisatio numerorum*" problem [20], and was further studied in the following years by Hille [15], Ikehara [19], Erdős [7], among others. To the best of our knowledge, interest in ordered factorizations not only for their counting function but as a *combinatorial object* traces back to Hwang's 1994 thesis [16].

More recently still, in 2005, Knopfmacher and Mays [22] wrote a survey showing renewed interest in various types of factorizations; interestingly, in [21], they approached *exhaustive generation* of ordered and unordered factorizations using algorithms mainly derived from known identities on the Dirichlet generating functions. But however clever these algorithms may be, they are intrinsically limited to relatively small factorizations.

With Dirichlet samplers introduced in this paper, it is possible to uniformly draw an ordered factorization of size about 10^{200} in a matter of seconds.

Building the sampler. In the context of the symbolic method, ordered factorizations can be described as the multiplicative class $\mathcal{F}$ specified by

$$\mathcal{F} := \textsc{Seq}\,(\mathcal{I} \setminus \mathcal{Z}_1)\,,$$

that is: an ordered factorization is a sequence of natural integers distinct from 1. The associated Dirichlet generating function is

$$F(s) := \frac{1}{1 - (\zeta(s) - 1)}.$$

Using the rules we have previously given on the construction of Dirichlet samplers, we can easily construct the sampler for class $\mathcal{F}$ which is given in Figure 4 (note for the purpose of clarity, the algorithm does not explicitly feature rejection).

The largest singularity of F is $\rho = \zeta^{(-1)}(2) \approx 1.72865$ (where $\zeta^{(-1)}(t)$ is the functional inverse of $\zeta(s)$, i.e. the real solution to $\zeta(s) = t$), and is a simple pole. Therefore, Theorem 2.2 applies and the control parameter to use, to generate ordered factorizations of size n with a logarithmic number of attempts, is

$$s_n = \zeta^{(-1)}(2) + \frac{1}{\log n}.$$

Some applications. We now have means by which to uniformly sample ordered factorizations of any given size; first order of business is to figure out what to do with that. It so happens Hwang has extensively studied the asymptotic distribution of *the number of factors* in random ordered factorizations in [17], and shown that it is a normal distribution[10]. Figure 5a shows empirical results validating Hwang's theorem.

But it is also easily possible to study other parameters of random ordered factorizations, such as: the number or (more trickily) the proportion of prime factors, of non prime factors; the number of different distinct factors; the (logarithm of) the maximum factor; the size of the longest "*run*" (sequence of consecutive equal factors). In Figure 5b, we have chosen to look at the number of 2s, 3s, 4s, etc. in a random ordered factorization and surprisingly it appears these may be *gamma*-distributed, which (if theoretically confirmed) would be an interesting property.

Boltzmann and Dirichlet samplers thus indeed have a high potential (which, as far as we know, has always been touted yet never fully taken advantage of, perhaps because these samplers remain too complex to implement) in helping draw conjectures on combinatorial objects—and in helping locate which results could be most interesting to prove.

3.2 Tiling Rectangles by Translation.

We now take a small break from number theory, to mention tilings of rectangle using a single tile which is copied and *translated*. As luck would have it, these tiles are described by a Dirichlet generating function: indeed, in [2], the authors showed that the number of tilings[11] for an interval of length n is in direct relation with the

[9] It is a rather unimportant detail, but for a given size n, Hwang considers the distribution of the factors of the factorizations *smaller than*, rather than *equal to* n. Thus why we generate a set of ordered factorizations of *random sizes between* 2 *and* 10^{20}.

[10] This result applies to the ordered factorizations which we have considered here, namely those with factors chosen from $\{2, 3, 4, 5 \ldots\}$; it was generalized by Hwang and Janson ten years later in [18], which extends it to ordered factorizations with factors chosen from any subset of integers.

[11] Since a tiling is the same tile which is translated all over the place, both can be considered the same object, and in this subsection we refer sometimes to one or the other.

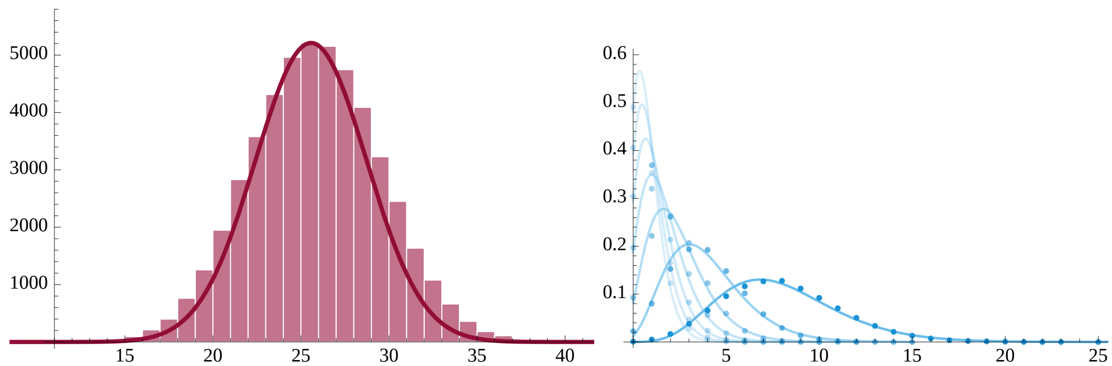

Figure 5. (a) Left: the histogram is the empirical distribution of the number of factors of about 50 000 ordered factorizations sampled from all ordered factorizations of sizes between 2 and 10^{20}, with a 10% error; super-imposed is the curve of the (appropriately scaled) theoretical normal law derived by Hwang[9] in [17]. (b) Right: 100 000 ordered factorizations of size 10^{20}, with a 10% error, have been generated; the dots are the empirical distributions of the 2s, 3s,...,6s, super-imposed with gamma curves which have been fitted to the empirical distributions; the most opaque dots-and-curve is for the distribution of 2s, the most transparent is for the distribution of 6s.

number of divisors of n. Their main theorem is stated here as Proposition 3.1.

Proposition 3.1. (Bodini & Rivals) *For $n \in \mathbb{Z}_{>0}$, let ξ_n be the number of tilings for an interval of length n, and let $R(s)$ be Dirichlet generating function of the tilings. Then*

$$\xi_1 = 1 \qquad \textit{and} \qquad \xi_n = 1 + \sum_{d|n,\, d\neq n} \xi_d$$

thus

$$R(s) := \sum_{n=1}^{\infty} \frac{\xi_n}{n^s} = \frac{\zeta(s)}{1-(\zeta(s)-1)}.$$

From the Dirichlet generating function $R(s)$ of the tiles, it is easy to see that these are in bijection with a special form of ordered factorization[12], which has at least one factor, and this factor may exceptionally be equal to 1. These sort of objects, as we have seen in the previous subsection, we know how to generate.

In short, two-dimensional tiles for a rectangle (of about) $n \times m$ can be obtained as follows: we draw two of those particular ordered factorization, one of size about n and one of size about m; we transform each of them using a bijection[13]; finally, we create a two-dimensional tile as the Cartesian product of both one-dimensional tiles. To obtain a tiling, we translate the tile as many times as needed to completely cover the rectangle.

[12] The recursive definition of regular ordered factorization is exactly the same as for these tiles, except the former's sum is over all divisors d of n that are distinct from n, *and from* 1.

[13] This bijection involves repeatedly applying a dual transformation to the tiles and growing them.

3.3 Multiplicative Trees.

Branching factorizations were introduced by Hwang in [16] as a general term to refer to the multiplicative analog of trees: the premise is to take some tree structure, label its leaves with elements from some subset of all integers, and consider the size of the tree to be the product of its leaves.

Though our generators are by no means limited to them, for simplicity's sake we will consider the class $\mathcal{T}$ of binary trees—the staple of combinatorialists!—with leaves any integer greater than 1. We abusively refer to these trees as "branching factorizations", and when we want to talk about all multiplicative tree-like structure, we use the slightly dull term "multiplicative trees".

Going through the motions. The symbolic specification of $\mathcal{T}$,

$$\mathcal{T} := \mathcal{I} \setminus \mathcal{Z}_1 + \mathcal{T} \times \mathcal{T}$$

translates to a quadratic equation on the Dirichlet generating function $T(s)$; once solved, it predictably appears that $T(s)$ does not have a pole, but a branch point at $\rho = \zeta^{(-1)}(5/4) \approx 2.78843$,

$$T(s) = \frac{1-\sqrt{5-4\zeta(s)}}{2}$$

and

$$T(s) \underset{s\to\rho}{\sim} \frac{1}{2} - \sqrt{\zeta'(\rho)} \cdot \sqrt{s-\rho}.$$

Theorem 2.3 applies here, thus the control parameter to use is

$$s_n = \zeta^{(-1)}(5/4) + \frac{\zeta'(\zeta^{(-1)}(5/4))^2}{(\log n)^2}.$$

This information is put to use in the thoroughly detailed Dirichlet sampler, complete with rejection, which is given in Figure 8. Since it takes an expected $\Theta((\log n)^{3/2})$ number of trials to draw an object within the desired size interval, and each object is generated in linear time, the complexity of the sampler is $\Theta((\log n)^{5/2})$—we will see this can be improved upon.

Many questions. Not much is known about these multiplicative trees. By applying the Delange transfer theorem to $T(s)$, Hwang was able to show (in the final chapter of his thesis, [16, 11.5]) that the number of factors/leaves of a random branching factorization is normally distributed[14]. This is a feature shared with a very large class of additive trees, as was shown by Meir and Moon in [23]. Furthermore, simulations in part described in Figure 7 show that the *shape* of the average profile of random branching factorizations seems to resemble that of random trees.

So on the one hand, some commonalities seem to indicate that these multiplicative trees could behave like additive trees. But on the other hand, there is evidence to the contrary: while the coefficients of additive trees are usually smooth, the coefficients of Dirichlet generating functions, among which those of multiplicative trees, are characteristically chaotic of the underlying importance of divisors, factorizations and primes; and the asymptotic of these coefficients, even when logarithmically scaled down, is very different from that of traditional combinatorial trees.

For instance, if we consider the class $\mathcal{A} := 2\mathcal{Z} \times \textsc{Seq}\,(2\mathcal{Z}) + \mathcal{A} \times \mathcal{A}$ (which for certain reasons, could resemble a multiplicative tree), the asymptotics obtained using the Flajolet-Odlyzko transfer theorem (see [9] or [11, VI]) and the Delange theorem, we obtain

$$\sum_{i=1}^{n} a_i \sim \frac{5\sqrt{5}}{36\sqrt{\pi}} 10^n n^{-3/2}$$

whereas

$$\sum_{i=1}^{n} t_i \sim \frac{\zeta'(\zeta^{(-1)}(5/4))}{\zeta^{(-1)}(5/4) \cdot \Gamma(-1/2)} n^{\zeta^{(-1)}(5/4)} (\log n)^{-3/2}.$$

Further investigation is warranted, as it would undoubtedly be very interesting to study various parameters of vast sets of very large randomly generated branching factorizations.

[14] This is unsurprising for many reasons; for instance, because of the close ties between the number of factors in ordered factorizations (which, as mentioned in Subsection 3.1, is normally distributed) and the number of leaves in a branching factorization.

4 Introducing a Dirichlet Sampler Algebra

As previously noted, those combinatorial classes to which Theorem 2.3 applies—for instance, tree structures such as the branching factorizations introduced in Subsection 3.3—take on average $\Theta((\log n)^{3/2})$ trials to sample an object, whereas in the general case only $\Theta(\log n)$ trials are needed.

This is analogous to the problem posed by the so-called 'peaked' distribution in the original Boltzmann paper [6]: objects with this distribution tend to produce many more smaller objects, to such an extent that larger instances prove very difficult to obtain. The authors' solution was to use the combinatorial *pointing* construction. This construction makes sizes more uniformly distributed, and thus makes it faster to sample larger objects. Unfortunately this solution cannot be applied as is to Dirichlet sampler because of a fundamental difference of multiplicative combinatorial classes.

Why differentiation is not pointing. The pointing operator takes an object and distinguishes (or differentiates) one of its atoms; applied to a whole class $\mathcal{B}$, the operator generates the class $\mathcal{A} = \Theta\mathcal{B}$, which contains every possible object which can result from distinguishing an atom from an object of $\mathcal{B}$.

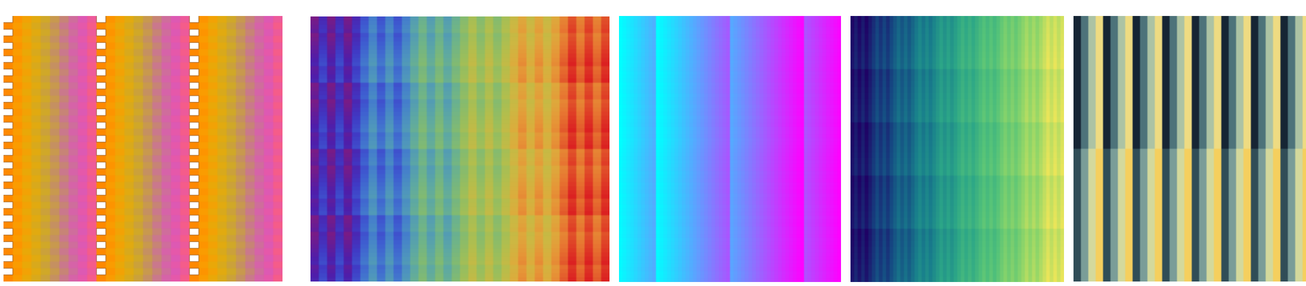

Figure 6. Tilings of variously sized rectangles (all scaled to the same height for aesthetic reasons), each *copy* of a tile is of a *different color*; for clarity, in the leftmost tiling, one of the copies of the tile stands out in white.

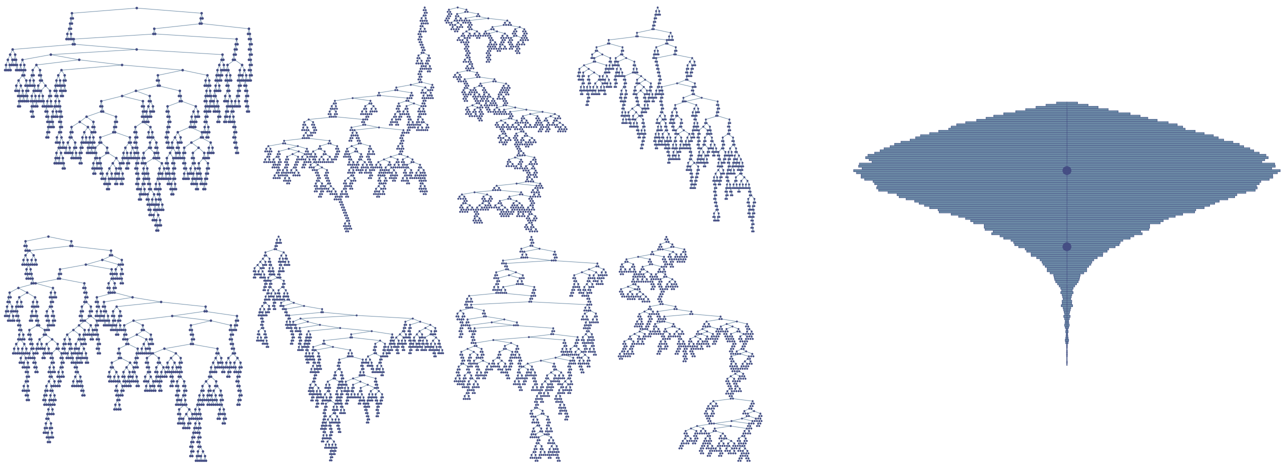

Figure 7. (a) Left: various shapes of branching factorizations of size 10^{200} with a 50% tolerance; obviously the actual factors, in the leaves, are *not* displayed—but you must imagine each leaf has a number and the product of all leaves is the size, so roughly 10^{200}. (b) Right: the average profile of 86 random branching factorizations from which the eight displayed trees are taken; the average profile is widest at the 33rd level, and the average height is 66 (both are marked by a dot).

```
global size

function ΓDBranchingFactorization(n, ε)
    s ← ζ^(−1)(5/4) + ζ′(ζ^(−1)(5/4))/log(n)^2
    p ← 2(ζ(s) − 1)/(1 − √(5 − 4ζ(s)))
    repeat
        size ← 1
        object ← BFAux(n, ε, s, p)
    until (1 − ε)n ⩽ size ⩽ (1 + ε)n
    return object
end function

function BFAux(n, ε, s, p)
    if size > (1 + ε)n then return 0
    if Ber(p) = 1 then
        repeat
            f ← ZetaLaw(s)
        until f ≠ 1
        size ← size · f
        return Leaf(f)
    else
        return Node(BFAux(n, ε, s, p),
                    BFAux(n, ε, s, p))
    end if
end function
```

Figure 8. A more concrete description of a Dirichlet sampler for the branching factorizations in Subsection 3.3, this features both normal rejection (to make sure the object returned by the auxiliary function, BFAux, is within the targeted size interval) and *anticipated rejection* (immediately stop all recursive calls to the auxiliary function if ever the generated object's size goes beyond the targeted interval) which is crucial, in Dirichlet samplers, to achieve the expected complexity.

In additive combinatorial structures, the *atomic class* $\mathcal{Z}$ happens to be *unique* and to have *unit size*. As a consequence, there is a direct correspondence between *the number of atoms* and *the size of an object*—they are one and the same. In this context, the pointing operator is agreeably translated to the generating functions (both ordinary and exponential) as a differentiation,

$$[z^n]A(z) = n \cdot [z^n]B(z) \qquad \text{thus} \qquad A(z) = z\frac{\mathrm{d}}{\mathrm{d}z}B(z).$$

None of this holds true with multiplicative combinatorial structures: atomic classes are plenty, since there is actually one per prime integer, and therefore size holds no direct relation to the number of atoms. As a consequence, differentiating a Dirichlet generating function

$$\frac{\mathrm{d}}{\mathrm{d}s}\sum_{n=1}^{\infty}\frac{a_n}{n^s} = -\sum_{n=1}^{\infty}\frac{a_n \log n}{n^s}$$

usually results in coefficients which are no longer integers[15]—and functions which no longer hold any combinatorial meaning. This neither means that pointing does not exist for multiplicative structures, nor that differentiation does not hold a significant interest, simply that they are no longer the same thing.

4.1 Pointing with Another Variable. Multiplicative classes do have an equivalent to the pointing operator. But this requires explicitly knowing the Dirichlet generating function (and better yet, the combinatorial specification of the class) and rewriting it as a bivariate Dirichlet generating function. Indeed, if we set

$$\zeta(s,u) := u\zeta(s)$$

(proceeding in a similar manner with the prime class, and individual atoms) and propagate this change upwards, then one can point structures by differentiating with respect to u.

4.2 An Algebra with Differentiation. Differentiation may not have a combinatorial translation, but that is not important. As we have stressed before, generating functions are only *tools* used by the Dirichlet samplers.

We introduce Dirichlet samplers of order 1, which we note $\Gamma\mathrm{D}_s^{[1]}[\mathcal{A}]$ as an algorithm that returns an object $\omega \in \mathcal{A}$ of size n with probability

$$\mathbb{P}_s[\omega] := -\frac{n^{-s}\log n}{A'(s)}. \tag{4.10}$$

The basic constructions for this algebra are given in Table 2. The correction proofs have been removed for space considerations.

[15] The obvious exception is when we limit ourselves, like for additive objects, to a unique atom $\mathcal{Z}_e$.

To illustrate how this works, we again look at branching factorizations specified as

$$\mathcal{T} := \mathcal{I} \setminus \mathcal{Z}_1 + \mathcal{T} \times \mathcal{T}.$$

The obvious Dirichlet sampler for this class would be:

$$\Gamma\mathrm{D}_s^{[0]}[\mathcal{T}] := \Gamma_s^0(\mathcal{I}\backslash\mathcal{Z}_1) + \Gamma\mathrm{D}_s^{[0]}[\mathcal{T}]^2.$$

This is the Dirichlet sampler which we have given in Subsection 3.3 and implemented in Figure 8; because the singular exponent of the class' associated DGF is $\alpha = -1/2$, Theorem 2.3 implies that the random sampling of these objects has a complexity of $\Theta((\log n)^{5/2})$. Now, if we differentiate the sampler, we obtain

$$\Gamma\mathrm{D}_s^{[1]}[\mathcal{T}] := \Gamma_s^1(\mathcal{I}\backslash\mathcal{Z}_1) + \Gamma\mathrm{D}_s^{[0]}[\mathcal{T}] \times \Gamma\mathrm{D}_s^{[1]}[\mathcal{T}] + \Gamma\mathrm{D}_s^{[1]}[\mathcal{T}] \times \Gamma\mathrm{D}_s^{[0]}[\mathcal{T}].$$

This sampler behaves as if the Dirichlet generating function had a singular exponent $\alpha = 1/2$, which yields a complexity of $\Theta((\log n)^2)$.

5 Conclusion

We have introduced Dirichlet random samplers, an automatic technique to sample from multiplicative combinatorial classes (so called because the Cartesian product's size is a product instead of a sum). Our samplers are efficient: indeed they can draw an object of approximate size n in $O((\log n)^2)$ worst-case time complexity. And this is fortunate as the size of multiplicative combinatorial objects grows very quickly!

To the best of our knowledge, while exhaustive generation of some specific multiplicative objects has already been approached, most recently by Knopfmacher and Mays [21], this seems to be the first time that random sampling of multiplicative objects has been considered in a generic way. By providing a means of quickly generating vast sets of large objects, our Dirichlet samplers can facilitate the investigation of properties of these interesting, yet notoriously hard to study objects. Dirichlet samplers could also be of interest from a number theoretic point of view.

But as a subtext, we also sought to convey much broader ideas regarding possible future directions for random generation.

Specifically, we believe that while there has been much interest in extending the expressiveness of Boltzmann sampling, little has been said about the main concept itself—obtaining probabilities from the evaluation of functions—and little has been done to further this concept[16].

[16] Pivoteau *et al.*'s work [26] is significant. But their primary goal is to optimize the Boltzmann method and make it effective for practical use, rather than extend the model and its core ideas.

Atoms	$\mathcal{A} = \mathcal{Z}_i$	$\Gamma\mathrm{D}_s^{[1]}[\mathcal{A}] := \square_i$
Union	$\mathcal{A} = \mathcal{B} + \mathcal{C}$	$\Gamma\mathrm{D}_s^{[1]}[\mathcal{A}] := \mathrm{Ber}\left(\dfrac{B'(s)}{A'(s)}\right) \longrightarrow \Gamma\mathrm{D}_s^{[1]}[\mathcal{B}] \mid \Gamma\mathrm{D}_s^{[1]}[\mathcal{C}]$
Product	$\mathcal{A} = \mathcal{B} \times \mathcal{C}$	$\Gamma\mathrm{D}_s^{[1]}[\mathcal{A}] := \mathrm{Ber}\left(\dfrac{B'(s) \cdot C(s)}{A'(s)}\right) \longrightarrow (\Gamma\mathrm{D}_s^{[1]}[\mathcal{B}], \Gamma\mathrm{D}_s^{[0]}[\mathcal{C}]) \mid (\Gamma\mathrm{D}_s^{[0]}[\mathcal{B}], \Gamma\mathrm{D}_s^{[1]}[\mathcal{C}])$

Table 2. Extended rules for the Dirichlet samplers of order 1; we use the shorthand notation introduced by [6], where $\mathrm{Ber}(p) \to A|B$ means "draw a Bernoulli of parameter p, if it succeeds (if it is equal to 1) return A, if not return B". Note that the Cartesian product, which used to be one of the only rules not to involve probabilities, here requires drawing a Bernoulli.

The use of Dirichlet generating functions, instead of ordinary or exponential generating functions, illustrates again that the generating function itself is, in the words of Herb Wilf, "*a clothesline on which we hang up a sequence of numbers for display*". This suggests that more work should be done to abstract away the specific kind of generating function (ordinary, exponential, Dirichlet, etc.) from the main concept (probabilities from evaluation), as the former seem to be red herrings which distract from the interesting theoretical developments that have yet to be done; in essence, we advocate a general model, "*Analytic Random Sampling*".

In a similar vein, the sampler algebra, which we have tentatively introduced in the last section as a means to improve the sampling complexity of tree-like structures, suggests a much more generic and systematic way of manipulating the generating functions to target size distributions—and perhaps reduce or remove the need for rejection, which does not seem to be an inevitability. This will be the subject of future work.

Acknowledgments

We warmly thank Hsien-Kuei Hwang for the helpful references he provided; and more generally, for introducing multiplicative combinatorial structures and writing about them. We are also very grateful to Yann Ponty for carefully reading our manuscript and providing us with thoughtful feedback.

This work was supported by the French ANR Magnum. The authors are appreciative of the referees of ANALCO-2012 for their attentive comments.

References

[1] Olivier Bodini and Yann Ponty. Multi-dimensional Boltzmann sampling of languages. *DMTCS Proceedings*, 0(01):49–64, 2010.

[2] Olivier Bodini and Eric Rivals. Tiling an interval of the discrete line. In *Combinatorial Pattern Matching*, pages 117–128. Springer, 2006.

[3] Hubert Delange. Généralisation du théorème de Ikehara. *Annales scientifiques de l'École Normale Supérieure Série 3*, 71(3):213–142, 1954.

[4] Luc Devroye. *Non-Uniform Random Variate Generation*. Springer Verlag, 1986.

[5] Philippe Duchon, Philippe Flajolet, Guy Louchard, and Gilles Schaeffer. Random sampling from Boltzmann principles. In Peter Widmayer et al., editor, *Automata, Languages, and Programming*, number 2380 in Lecture Notes in Computer Science, pages 501–513. Springer Verlag, 2002.

[6] Philippe Duchon, Philippe Flajolet, Guy Louchard, and Gilles Schaeffer. Boltzmann samplers for the random generation of combinatorial structures. *Combinatorics, Probability and Computing*, 13(4–5):577–625, 2004. Special issue on Analysis of Algorithms.

[7] Paul Erdős. On some asymptotic formulas in the theory of the "Factorisatio Numerorum". *The Annals of Mathematics*, 42(4):989–993, 1941.

[8] Philippe Flajolet, Éric Fusy, and Carine Pivoteau. Boltzmann sampling of unlabelled structures. In David Applegate *et al.*, editor, *Proceedings of the Ninth Workshop on Algorithm Engineering and Experiments and the Fourth Workshop on Analytic Algorithmics and Combinatorics*, pages 201–211. SIAM Press, 2007. Proceedings of the New Orleans Conference.

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

[10] Philippe Flajolet, Maryse Pelletier, and Michèle Soria. On Buffon machines and numbers. In Dana Randall, editor, *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011*, pages 172–183. SIAM, 2011.

[11] Philippe Flajolet and Robert Sedgewick. *Analytic Combinatorics*. Cambridge University Press, 2009. 824 pages (ISBN-13: 9780521898065); also available electronically from the authors' home pages.

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

[13] Carl-Erik Fröberg. On the prime zeta function. *BIT*

Numerical Mathematics, 8(3):187–202, 1968.

[14] Éric Fusy. Uniform random sampling of planar graphs in linear time. *Random Structures & Algorithms*, 35(4):464–522, 2009.

[15] Eínar Hille. A problem in "Factorisatio Numerorum". *Acta Arithmetica*, 2(1):134–144, 1936.

[16] Hsien-Kuei Hwang. *Théorèmes limites pour les structures combinatoires et les fonctions arithmetiques.* Ph.D. Thesis, École Polytechnique, December 1994.

[17] Hsien-Kuei Hwang. Distribution of the Number of Factors in Random Ordered Factorizations of Integers. *Journal of Number Theory*, 81(1):61–92, September 1999.

[18] Hsien-Kuei Hwang and Svante Janson. A central limit theorem for random ordered factorizations of integers. *Electronic Journal of Probability*, 16:347–361, 2011.

[19] Shikao Ikehara. On Kalmar's problem in "Factorisatio Numerorum". *Journal of the Physico-Mathematical Society of Japan*, 21:208–219, 1939.

[20] László Kalmár. A "factorisatio numerorum" problémájáról. *Matematikai és Fizikai Lapok*, 38:1–15, 1931.

[21] Arnold Knopfmacher and Michael E. Mays. Ordered and unordered factorizations of integers. *Mathematica Journal*, 10(1):72–89, 2005.

[22] Arnold Knopfmacher and Michael E. Mays. A survey of factorization counting functions. *International Journal Number Theory*, 1(4):563–581, 2005.

[23] Amram Meir and John W. Moon. On nodes of given out-degree in random trees. In Jaroslav Nešetřil and Miroslav Fiedler, editors, *Fourth Czechoslovakian Symposium on Combinatorics, Graphs and Complexity*, volume 51 of *Annals of Discrete Mathematics*, pages 213–222. Elsevier, 1992.

[24] Albert Nijenhuis and Herbert S. Wilf. *Combinatorial Algorithms.* Academic Press, 2nd edition, 1978.

[25] Carine Pivoteau, Bruno Salvy, and Michèle Soria. Boltzmann oracle for combinatorial systems. In *Proceedings of the Fifth Colloquium on Mathematics and Computer Science. Blaubeuren, Germany*, pages 475–488, 2008.

[26] Carine Pivoteau, Bruno Salvy, and Michèle Soria. Algorithms for combinatorial systems, 2011, arXiv:1109.2688v1.

[27] Gérald Tenenbaum. *Introduction to analytic and probabilistic number theory.* Cambridge University Press, 1995. Translated from the second French edition (1995) by Charles B. Thomas.