Analytic Samplers and the Combinatorial Rejection Method

Olivier Bodini Jérémie Lumbroso Nicolas Rolin

September 9, 2014

Abstract

Boltzmann samplers, introduced by Duchon *et al.* in 2001, make it possible to uniformly draw approximate size objects from any class which can be specified through the symbolic method. This, through by evaluating the associated generating functions to obtain the correct branching probabilities.

Evaluating generating function (in particular in the neighborhood of their singularity), is generally a complex problem, which long remained an unsolved aspect of the method. At first these computations were usually outsourced to a computer algebra system. But more recently, in 2008, Pivoteau *et al.* brought a sweeping answer to the question by introducing an efficient iteration method, which could evaluate any function system with a combinatorial origin with quadratic convergence.

In both cases though, the evaluations—though they can usually be obtained with arbitrary precision—are often obtained with truncated accuracy, perhaps to 20 digits, because it is considered that the impact on uniformity is negligeable. Furthermore, though the question of evaluating the generating is solved, the question of precisely calibrating the samplers for a fixed size is still partially unresolved.

By adapting the rejection method, a classical tool from the random, we show how to obtain a variant of the Boltzmann sampler framework, which is tolerant of approximation, even large ones. Our goal for this is twofold: this allows for exact sampling with approximate values; but this also allows much more flexibility in tuning samplers. For the class of simple trees, we will try to show how this could be used to more easily calibrate samplers.

1 Introduction

Being able to randomly generate large combinatorial objects of any given class, is a fundamental problem with countless applications in scientific modeling.

Nijenhuis and Wilf introduced the *recursive method* [14] in the late 70s (later extended by [11]), 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 [6, 7], 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 counting generating functions, that is for an unlabelled combinatorial class $\mathcal{C}$, for which there are c_n elements of size n, its generating function is defined as

$$C(z) := \sum_{n=1}^{\infty} c_n z^n.$$

Through evaluation, all the coefficients of a generating function are smashed together, and the resulting probabilities take into account objects of *all sizes*. Thus, while 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: rejecting objects which are not within the desired size interval; 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—in many regards Boltzmann sampling is an elegant and natural application of *Analytic Combinatorics* pioneered by Flajolet and Sedgewick [10]—have made it a fertile topic, and many of its aspects have been developed through many papers.

The Boltzmann model. With a Boltzmann sampler for an unlabelled combinatorial class $\mathcal{C}$, for which there are c_n elements of size n, the probability of drawing an object $\gamma \in \mathcal{C}$ is

$$\mathbb{P}_z[\gamma] = \frac{z^{|\gamma|}}{C(z)} \qquad \text{with} \qquad C(z) := \sum_{n=1}^{\infty} c_n z^n = \sum_{\gamma \in \mathcal{C}} z^{|\gamma|}$$

where $|\gamma|$ denotes the size of object γ and x is some control parameter to be chosen. Thus the probability of drawing an object of size n is

$$\mathbb{P}_z[|\gamma| = n] = \frac{c_n z^n}{C(z)} \qquad\qquad \mathbb{P}_s[\gamma \mid |\gamma| = n] = \frac{1}{c_n}$$

while the probability of drawing an object conditioned on its size is uniform.

The name of the method evokes the Boltzmann model of statistical physics that assigns to each possible state of a system probability $\mathrm{e}^{-\beta E}/Z$, where E is the energy of the state, $\beta = 1/T$ is a constant, and Z is normalizing constant. Though the distribution of the sizes of objects is a very generic distribution already known to probabilists as the *Power Series Distribution*[1], and according to Nelson *et al.* [12, §2.2], the terminology is usually credited to Noack [15] in 1950.

Evaluating generating functions near their singularity. Boltzmann samplers depend on the evaluation of generating function in the neighborhood of their singularity—what more in constant time arithmetic complexity.

The problem of how these functions should be evaluated was left open by the original paper, and was without answer until the contribution of Pivoteau *et al.* [17, 18]. They introduced a variant of Newton's iteration for combinatorial systems, which has a highly efficient quadratic convergence.

However because of the finite nature of computers, although the oracle can provide arbitrary precise values, we usually restrain ourselves to fixed precision approximations. It has been argued that the incurred bias in uniformity is minimal. This is true, but with our method we provide a way to have exact simulations.

In addition, solving the evaluation problem does not entirely resolve the issue of tuning the samplers: that is, picking the value of z, which will yield the best concentration of objects of the targeted size. Currently, expected value tuning requires inverting a system of equations, and to our knowledge this is not routinely done for large combinatorial systems. For certain combinatorial classes (algebraic classes), singular samplers are tuned by approaching the singularity as close as possible: this requires making a logarithmic number of calls to the oracle [4].

1.1 Our contribution: an extended, practical framework

Our idea appeals to the classical random generation concept of *rejection* (see for instance Devroye's chapter on the rejection method [5, §2]). Instead of evaluating the generating functions exactly, we pick some nearby point that is easier to computer.

[1]The Poisson, geometric, log-series distributions are all special cases of this distributions, a fact which is put to use by Flajolet *et al.* [9, §2] who designed the Von Neumann/Flajolet scheme to simulate power series distributions using only random bits.

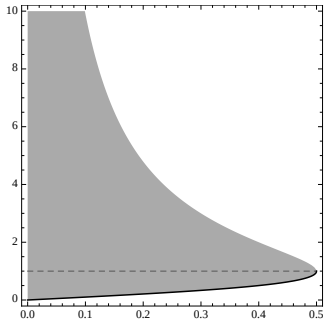

Figure 1. Plot associated with the combinatorial specification of binary trees where all nodes are counted, $\mathcal{B} = \mathcal{Z} + \mathcal{Z} \times \mathcal{B}^2$. The thick black curve at the bottom plots $C(z) = z + zC(z)^2$, or all coordinates $(z, C(z))$ usually considered for Boltzmann sampling; the shaded area is the region verifying $c \geqslant z + zc^2$, and from which we get coordinates (z, c) which we use in our modified model.

This is illustrated in Figure 1. Both Boltzmann samplers and our samplers use coordinates from the shaded region. But while Boltzmann samplers limit themselves to the coordinates that belong to the thick black curve at the bottom of the region, we allow ourselves to pick any point within the region. Of course, this comes at the cost of additional rejection, but we show that this rejection is constant and that for reasonable choices of coordinates it is practically negligible.

By not restricting ourselves to a fixed curve, we are given some latitude with which to pick the points. For instance, rational numbers yield probabilities that are much easier to simulate exactly and efficiently [**?**], and are easier to manipulate exactly (notably when using dichotomic search). Since rational numbers are dense in real numbers, it will prove very useful to only use rational numbers in our samplers, using a Farey sequence approximation—and this is a freedom of action that is afforded us by our use of rejection.

The ideas presented here followed from the first author's work to extend Boltzmann samplers to infinite objects [2], and the second author's attempts to modify the generating functions to shape the size distribution of the sampled objects.

2 Analytic Samplers

In this section we give the main definitions for our analytic random samplers, and then show the algorithms associated with the basic constructions.

2.1 Main definitions

DEFINITION 1. *Let $\mathcal{A}$ be an unlabelled combinatorial class, and a_n the number of objects from $\mathcal{A}$ that have size n. The* ordinary generating function *(OGF) associated with class $\mathcal{A}$ is defined equivalently by*

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

The ordinary generating function enumerates combinatorial class it is associated to. The tenet of the *symbolic method* [10] is that if a combinatorial class can be symbolically specified using a set of operators (disjoint union, Cartesian product, sequence, multiset, etc.) from initial terminal symbols called *atoms* which have unit size, then this specification can be directly translated to obtain the ordinary generating function.

DEFINITION 2. *Let $\mathcal{A}$ be a symbolically defined combinatorial class which can be translated, following the symbolic method, to a functional equation on $A(z)$, the generating function associated with $\mathcal{A}$,*

$$\mathcal{A} = \Phi(\mathcal{Z}, \mathcal{A}, \mathfrak{X}) \qquad \Rightarrow \qquad A(z) = \phi(z, A(z), \boldsymbol{X}(z)),$$

where both Φ and ϕ may possibly involve other classes/generating functions which we note using vectors in bold (and each symbol/generating function component of the vector itself defined by their own equations). A pair of coordinates (z, a) is said to be analytically valid coordinates *for the combinatorial class $\mathcal{A}$ if and only if they verify the inequality*

$$a \geqslant \phi(z, a, \boldsymbol{x}).$$

In general, for convenience and clarity, we will omit the vector in the notations, and any additional bound symbols will be implicit.

DEFINITION 3. *An* analytic sampler *for an unlabelled combinatorial class $\mathcal{A}$ is an algorithm which samples an object $\alpha \in \mathcal{A}$, of size $|\alpha|$, with probability*

$$\mathbb{P}_{(z,a)}[\alpha] = \frac{z^{|\alpha|}}{a} \qquad \textit{and fails with probability} \qquad \mathbb{P}_{(z,a)}[\dagger] = 1 - \frac{A(z)}{a}$$

where $A(z)$ is the ordinary generating function associated with class $\mathcal{A}$, and the analytically valid coordinates (z, a) are called the control parameter. *Moreover we denote by $\Gamma\mathcal{A}(z, a)$ such an analytic sampler.*

Because the original Boltzmann samplers already used the concept of rejection to control the size of the output, and constrain it to a tolerance interval, we choose instead to call our additional rejection, *death*, to avoid confusion.

Theorem 1. *Let $\mathcal{A}$ be a combinatorial class and $A(z)$ its generating function, and let (z, a) be analytically valid coordinates for $\mathcal{A}$. The proportion of objects of which the generation has failed does not depend on the size of the finally output object, and is equal to $A(z)/a$.*

Proof. This follows from the definition of the model of Analytic Samplers wherein the probability of a single draw failing is constant—in the sense that it does not depend on the size of the object that was being constructed when the sampling failed—and equal to

$$\mathbb{P}_{(z,a)}[\dagger] = 1 - \frac{A(z)}{a},$$

and thus an object (of some random size) is drawn with the complementary probability. The number of failures before an actual object is drawn is then geometrically distributed with $p = 1 - A(z)/a$. We then have:

$$\begin{aligned}
\mathbb{E}_{(z,a)}[\#\dagger] &= \sum_{k=0}^{\infty} n \left(1 - \frac{A(z)}{a}\right)^k \frac{A(z)}{a} \\
&= \frac{A(z)}{a} \frac{\left(1 - \frac{A(z)}{a}\right)}{\left(\frac{A(z)}{a}\right)^2} \\
&= \frac{a}{A(z)} \left(1 - \frac{A(z)}{a}\right) \\
&= \frac{a}{A(z)} - 1
\end{aligned}$$

and because there is one last object generated (the one that does not fail, and after which we are done) the expected proportion of objects which have failed is $1/(\mathbb{E}_{(z,a)}[\#\dagger] + 1)$ as stated. □

Note that the lower bound for a is $a = A(z)$, and for this choice of value, the analytic sampler does not fail[2]; indeed the inequality can naturally be seen as an equality involving a slack variable δ, $a = \phi(z, a, \boldsymbol{x}) + \delta$, where δ is the proportion of failures. In essence, if you are willing to spend the computational time needed to compute the generating function, then you are rewarded for your efforts by having no rejection at all.

Example 1. At this point we will illustrate these definitions by looking at the class $\mathcal{B}$ of binary trees, in which all nodes both internal and external count towards the size of the tree. These trees can be symbolically specified as either a leaf ($\mathcal{Z}$) or node which has two subtrees ($\mathcal{Z} \times \mathcal{B}^2$),

$$\mathcal{B} = \mathcal{Z} + \mathcal{Z} \times \mathcal{B}^2 \qquad \text{and} \qquad B(z) = z + zB(z)^2. \tag{1}$$

This functional equation can then be translated to an inequality,

$$b \geqslant z + zb^2.$$

The analytically valid coordinates for $\mathcal{B}$ are all points that belong to the shaded region in Figure 1.

2.2 Constructions

In this subsection, we give the basic constructions used by our analytic samplers. We follow the notation of the original article [7], which we extend it to include our failure probability,

$$\Gamma\mathcal{A} : [p_1] \cdot \mathrm{Ber}(p_2) \Rightarrow X \mid Y$$

means that we first fail with probability $1 - p_1$, then we draw a Bernoulli variable U of parameter p_2, if it is equal $U = 1$ then we return X if not we return Y. We instead of a Bernoulli distribution, we have a discrete distribution K, we mean that we return a tuplet of K independent calls to the sampler.

Let $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ be combinatorial classes. We recall, for clarity, that in this article we note $\mathbb{P}[\alpha]$ the probability of drawing an object α; when we want to make explicit from which class this object is drawn, we note $\mathbb{P}[\alpha \in \mathcal{A}]$.

Disjoint union. Let $\mathcal{A} = \mathcal{B} + \mathcal{C}$, and $a \geqslant b + c$. We first reject $1 - (b + c)/a$ of the objects, then we do a normal Boltzmann.

$$\Gamma\mathcal{A} : \left[\frac{b+c}{a}\right] \cdot \mathrm{Ber}\left(\frac{b}{b+c}\right) \Rightarrow \Gamma\mathcal{B} \mid \Gamma\mathcal{C} \tag{2}$$

Proof. We must show that the sampler $\Gamma\mathcal{A}$ returns objects $\alpha \in \mathcal{A}$ with the correct probability $\mathbb{P}_{(z,a)}[\alpha] = z^{|\alpha|}/a$ (that is the probability of drawing an object from $\mathcal{A}$ follows the law given in Definition 2), assuming inductively that the generators $\Gamma\mathcal{B}$ and $\Gamma\mathcal{C}$ are correct.

Hence:

$$\mathbb{P}_{(z,a)}[\alpha \in \mathcal{A}] = \frac{b+c}{a}\left(\frac{b}{b+c}\mathbb{P}_{(z,b)}[\alpha \in \mathcal{B}] + \frac{c}{b+c}\mathbb{P}_{(z,c)}[\alpha \in \mathcal{C}]\right)$$

That is the probability of sampling an object from $\mathcal{A}$ is the probability of first not failing, $(b + c)/a$, and then the probability of drawing the object using the sampler for class $\mathcal{B}$ or $\mathcal{C}$ with the correct Bernoulli probability. By hypothesis those two samplers return objects with correct probability, so

$$\mathbb{P}_{(z,a)}[\alpha \in \mathcal{A}] = \frac{b+c}{a}\left(\frac{b}{b+c}\frac{z^{|\alpha|}}{b} + \frac{c}{b+c}\frac{z^{|\alpha|}}{c}\right) = \frac{z^{|\alpha|}}{a}.$$

Note that in this proof, and the following, we do not explicitly prove the probability of failure as it is a straightforward consequence: sum the probability of drawing an object over all possible possible objects, and take the complimentary probability. □

[2]Because Boltzmann sampler/analytic sampler already involve some kind of rejection to target the size, we call this second type of rejection "failure" to avoid any confusion.

Cartesian product. Let $\mathcal{A} = \mathcal{B} \times \mathcal{C}$, and $a \geqslant b \cdot c$.

$$\Gamma\mathcal{A} : \left[\frac{b \cdot c}{a}\right] \Rightarrow (\Gamma\mathcal{B} \ ; \ \Gamma\mathcal{C}) \tag{3}$$

Proof. The proof follows the same model as the previous construction; let $\alpha = (\beta, \gamma)$,

$$\mathbb{P}_{(z,a)}[\alpha \in \mathcal{A}] = \frac{b \cdot c}{a} \mathbb{P}_{(z,b)}[\beta \in \mathcal{B}] \, \mathbb{P}_{(z,c)}[\gamma \in \mathcal{C}]$$

and since the samplers for $\Gamma\mathcal{B}$ and $\Gamma\mathcal{C}$ are inductively assumed to be correct,

$$\mathbb{P}_{(z,a)}[\alpha \in \mathcal{A}] = \frac{b \cdot c}{a} \frac{z^{|\beta|}}{b} \frac{z^{|\gamma|}}{c} = \frac{z^{|\beta|+|\gamma|}}{a} = \frac{z^{|\alpha|}}{a}.$$

□

Sequence. Let $\mathcal{A} = \textsc{Seq}\,(\mathcal{B})$ and $a \geqslant 1/(1-b)$.

$$\Gamma\mathcal{A} : \left[\frac{(1-b)^{-1}}{a}\right] \cdot \mathrm{Geo}(b) \Rightarrow (\Gamma\mathcal{B}, \ldots) \tag{4}$$

Proof. We follow the same model as previously. Let $\alpha = (\beta_1, \ldots, \beta_k)$.

$$\mathbb{P}_{(z,a)}[\alpha \in \mathcal{A}] = \frac{(1-b)^{-1}}{a} \mathbb{P}[\mathrm{Geo}(b) = k] \prod_{i=1}^{k} \mathbb{P}_{(z,b)}[\beta_i \in \mathcal{B}]$$

and by hypothesis

$$\begin{aligned}
\mathbb{P}_{(z,a)}[\alpha \in \mathcal{A}] &= \frac{(1-b)^{-1}}{a} b^k (1-b) \prod_{i=1}^{k} \frac{z^{|\beta_i|}}{b} \\
&= \frac{(1-b)^{-1}}{a} b^k (1-b) \frac{z^{|\beta_1|+\ldots+|\beta_k|}}{b^k} \\
&= \frac{z^{|\beta_1|+\ldots+|\beta_k|}}{a} = \frac{z^{|\alpha|}}{a}.
\end{aligned}$$

□

2.3 Illustration of "Death" Rate with Cayley Trees

For the purpose of giving an example that is somewhat more interesting than regular trees, we go slightly beyond the scope of unlabelled constructions which we have presented thus far, and venture into a labelled class which uses the SET operator. The point of this subsection is to illustrate on an example how moving away from the curve of a generating function impacts the death rate—that is, the rejection which must be done to compensate for the approximation.

Consider the example given by the class $\mathcal{T}$ of Cayley trees (labelled, unrestricted, non-plane trees), symbolically specified as $\mathcal{T} = \mathcal{Z} \star \textsc{Set}\,(\mathcal{T})$. Its exponential generating function, $T(z) = z\mathrm{e}^{-T(z)}$, is closely related to Lambert's W-function, which is implicitly defined. Actual standalone[3] implementations of Boltzmann samplers requiring this function have, for instance, resorted to using its truncated Taylor series expansion, see Bassino *et al.* [1].

[3] By *standalone*, we are referring to Boltzmann samplers not implemented within a computer algebra system, such as Maple or Mathematica, which usually provide computational access to such functions.

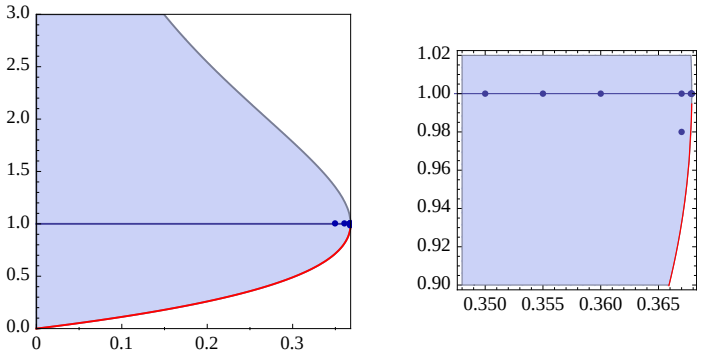

Figure 2. This a plot of the region defined by the inequality $t \geqslant z \cdot \exp(t)$, with z on the x-axis and t on the y-axis. The lower bound of the region, in bolded-red, is the curve of the exponential generating function $T(z)$. Each point corresponds to one of the columns of Table 1. The second figure, on the right, is a close-up near the singularity, at $z = \mathrm{e}^{-1}$.

With analytic samplers, our starting point is the system of functional equation yielded by the symbolic method (here there is only a single equation), replace any occurrence of a function by a free variable, and obtain the inequality $t \geqslant z \cdot \exp(t)$ and the algorithm

$$\Gamma\mathcal{T}(z,t) : \left[\frac{z\exp(t)}{t}\right] \cdot \mathrm{Poi}(t) \Rightarrow \square(\Gamma\mathcal{T}(z,t), \ldots, \Gamma\mathcal{T}(z,t))$$

that is, after an initial rejection (what we call death) with probability $z \cdot \exp(t)/t$ to account for the approximation of the generating function, we draw a Poisson variate of parameter t, to indicate how many children to generate.

It is straightforward enough to see that this algorithm is correct for any pair (z,t) which satisfies the aforementioned inequality. Notice also the rejection ratio can easily be simulated exactly using techniques described by Flajolet *et al.* [9].

More remarkably, an experiment summarized in Table 1 illustrates that the impact of approximation is modest. For various pairs of (z,t), the table summarizes the result of making 1000 calls to the sampler: it indicates in what proportion the sampling failed prematurely; and makes note of the average and maximal size among the trees actually drawn. The case where $z = \mathrm{e}^{-1}$ and $t = 1 = T(z)$ is special: first because this is the only case in which t is exactly equal to the evaluated EGF (thus we are 0% death and are analytic sampler is a traditional Boltzmann sampler); second because since we are evaluating the EGF in its singularity, this is actual a *singular Boltzmann sampler* (for which the expected value of the size of the output in unbounded). All other points, as illustrated in Figure 2 are more or less distant to the plot of $T(z)$, with a consequently higher death rate: but even at relatively significant distance from the curve, the death rate remains largely tolerable.

3 Simply Generated Trees

We now would like to illustrate how dealing with a region (and inequation) might make searching for an optimal pair of values for the sampler easier.

Simply generated trees were introduced by Meir and Moon [13] as classes of trees defined by the following specification

$$\mathcal{Y} = \mathcal{Z} \times \Phi(\mathcal{Y}) \tag{5}$$

	$t=1$							$t=0.98$
$z \;=$	0.35	0.36	0.367	0.3678	0.36787	0.367879	e^{-1}	0.367
death (observed)	28.8%	19.2%	6.4%	1.7%	0.4%	0.3%	0%	3.5%
death (theoretical)	28.3%	19.4%	6.8%	2.1%	0.7%	0.2%	0%	3.9%
average size	6.6	9.9	28.8	127.	177.3	2716.7	4944.3	35.9
maximal size	235	131	1493	17 799	26 531	826 167	2 518 975	1563

Table 1. This table summarizes the result of making 1000 calls to an analytic sampler for Cayley trees, with various values of z (the control parameter) and t (the approximation of the generating function). The *death* is the ratio of trees that must be rejected as a direct result of approximating the generating function, instead of evaluating it. Thus for the pair of values $z = \mathrm{e}^{-1}$ and $t = 1 = T(z)$, in which we use the exact value of $T(z)$, our samplers are exactly equivalent to Boltzmann samplers, hence the death is of 0%. What is remarkable is that the death resulting from rather large approximation remains manageable.

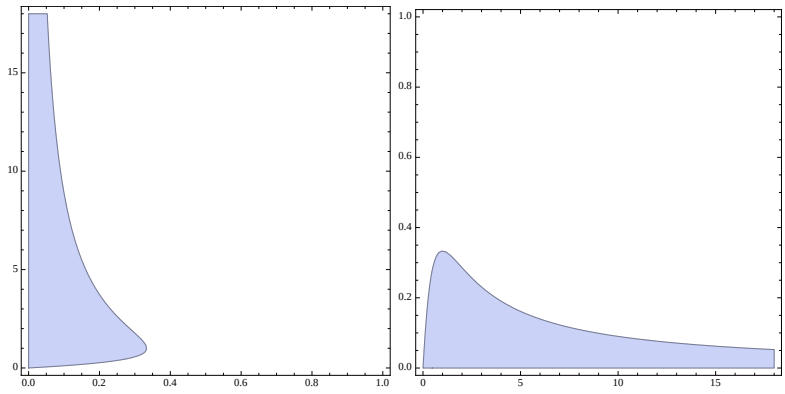

Figure 3. This is the plot of a the generating function of a simply generated tree (in this case, the class $\mathcal{U}$ of unary-binary trees, i.e., $\Omega = \{0, 1, 2\}$). On the left, the x-axis is z, the control parameter and the y-axis has $U(z) = z\Phi(U(z))$. On the right, we are plotting the function $u \mapsto u/\Phi(u)$. The problem of looking for the singularity, left, has been reduced to the more palatable problem of maximizing a function, right.

where Φ is a polynomial defined as

$$\Phi(w) = \sum_{\omega \in \Omega} w^\omega \qquad \Phi(w) = \sum_{\omega \in \Omega} \frac{w^\omega}{\omega!} \tag{6}$$

respectively depending on whether the class is unlabelled or labelled, and where $\Omega \subseteq \mathbb{N}$ is the multiset of allowable degrees (for instance, for binary trees, $\Omega = \{0, 2\}$). Meir and Moon identified that trees families defined in such a way shared an important number of common properties (such as mean path length of order $n\sqrt{n}$ or average height of order $\sqrt{n}$).

3.1 Existing Approaches

Randomly sampling from this class of tree is no longer particularly challenging: there are several methods to do this, with various properties of optimality (time, random-bit, etc.). So we do not presume to introduce samplers with any sort of new efficiency. However the example of simply generated introduce illustrates a way in which calibration might be more practical with analytic samplers.

Simply generated trees happen to have a branching singularity. This means: that their generating function can be evaluated at the singularity, and also that the size distribution of objects produced by a Boltzmann

sampler would be 'peaked', that is, highly concentrated towards smaller objects. The solution has traditionally been to do singular sampling: to pick z as being at, or near, the singularity, generate objects with unbounded expected size, and reject those that are too big.

Except in simple cases (such as binary trees, for which the singularity is well known to be $1/4$), the singularity is *not* known, so it must be determined empirically. This is usually done with a binary search, as implemented by Darrasse [4]: the oracle introduced by Pivoteau *et al.* [18] converges when inside the radius of convergence, and diverges otherwise; thus it is possible to determine whether we have gone over the singularity. This method requires a logarithmic number of calls to the oracle—a logarithmic number of evaluations that are not done in constant time.

3.2 Analytic Samplers: Maximizing a Polynomial

From the specification in Equation 5, we obtain the condition for analytic-validity of a pair (z, y),

$$y \geqslant z \cdot \Phi(y). \tag{7}$$

With this, it is now easier to not look at the generating function $Y(z)$, but instead at the function $y \mapsto y/\Phi(y)$, which is a rational function. This function admits a maximal point in the unit interval, which is the singular point of $Y(z)$.

Looking for this maximal point is a considerably easier problem, that does not require any evaluation of the generating function (except perhaps for a first guess): it can be solved by differentiation, by Newton iteration, or with a specifically optimized algorithm available in the litterature, such as Brent's algorithm [3].

4 Substitution Operator

While we have only described, for space and pertinence purposes, how to build analytic samplers for classes using elementary constructors, the possibilities are much broader. In particular, functional operators, such as pointing (differentiation) or substituting (composition) can naturally be used.

We will not go into detail, but instead provide the example of the unordered pair, $\textsc{MSet}_2$, and present an application with the random sampling of Otter trees.

4.1 Unordered Pair Construction

Let $\mathcal{B}$ be a combinatorial class, and $\mathcal{A} = \textsc{MSet}_2(\mathcal{B})$ be the class containing unordered pairs of elements of $\mathcal{B}$. The corresponding generating functions $A(z)$ and $B(z)$ verify the functional equation

$$A(z) = \frac{B(z)^2 + B(z^2)}{2}. \tag{8}$$

Assuming there is an analytic sampler for $\mathcal{B}$, we can build an analytic sampler for $\mathcal{A}$. Let (z, b) and $(z^2, \overline{b})$ both be analytically valid for $\mathcal{B}$ (note that the variable z must be the same in both pairs), and let (z, a) be analytically valid for $\mathcal{A}$, that is

$$a \geqslant \frac{b^2 + \overline{b}}{2}. \tag{9}$$

Using the notation we have introduced,

$$\Gamma\mathcal{A}(z,a) : \left[\frac{b^2+\overline{b}}{2a}\right] \cdot \mathrm{Ber}\left(\frac{b^2}{b^2+\overline{b}}\right) \Rightarrow \{\Gamma\mathcal{B}(z,b)\ ;\ \Gamma\mathcal{B}(z,b)\} \ \mid\ \Gamma\mathcal{B}(z^2,\overline{b}) \text{ and duplicate.} \tag{10}$$

In other terms, after making the obligatory failure test, we choose with the proper probability whether to create a pair of elements resulting from independent calls to $\Gamma\mathcal{B}(z, b)$, or whether to make one call to $\Gamma\mathcal{B}(z^2, \overline{b})$ and duplicating the resulting object to make a pair of identical objects.

Proof. As before, proving the validity of this algorithm involves showing that the analytic sampler $\Gamma\mathcal{A}(z, a)$ returns any object $\alpha \in \mathcal{A}$ with probability $z^{|\alpha|}/a$. We distinguish two disjoint cases.

- Either the pair $\alpha = \{\beta_1; \beta_2\}$ contains two distinct elements, $\beta_1 \neq \beta_2$. Then this pair could only have been produced by two independent (and distinguished) calls to $\Gamma\mathcal{B}(z, b)$. Thus under this setting,

$$\mathbb{P}_{z,a}[\alpha \mid \beta_1 \neq \beta_2] = \frac{b^2 + \overline{b}}{2a} \cdot \frac{b^2}{b^2 + \overline{b}} \left(\mathbb{P}_{z,b}[\beta_1] \cdot \mathbb{P}_{z,b}[\beta_2] + \mathbb{P}_{z,b}[\beta_2] \cdot \mathbb{P}_{z,b}[\beta_1]\right). \tag{11}$$

  By hypothesis, $\Gamma\mathcal{B}(z, b)$ is an analytic sampler for class $\mathcal{B}$, which means it returns an object $\beta \in \mathcal{B}$ with probability $z^{|\beta|}/b$,

$$\mathbb{P}_{z,a}[\alpha \mid \beta_1 \neq \beta_2] = \frac{b^2 + \overline{b}}{2a} \cdot \frac{b^2}{b^2 + \overline{b}} \frac{2z^{|\beta_1|+|\beta_2|}}{b^2} = \frac{z^{|\beta_1|+|\beta_2|}}{a} = \frac{z^{|\alpha|}}{a}. \tag{12}$$

- Or the pair $\alpha = \{\beta; \beta\}$ contains two identical objects. The pair could then have been drawn by either branch: from two independent calls to $\Gamma\mathcal{B}(z, b)$ which happen to return the same object; or from the call to $\Gamma\mathcal{B}(z^2, \overline{b})$ which is duplicated. In this case,

$$\mathbb{P}_{z,a}[\alpha \mid \beta = \beta] = \frac{b^2 + \overline{b}}{2a} \cdot \left(\frac{b^2}{b^2 + \overline{b}} \cdot \mathbb{P}_{z,b}[\beta]^2 + \frac{b^2}{b^2 + \overline{b}} \cdot \mathbb{P}_{z^2,\overline{b}}[\beta]\right). \tag{13}$$

  Again assuming the analytic sampler for $\mathcal{B}$ is correct,

$$\mathbb{P}_{z,a}[\alpha \mid \beta = \beta] = \frac{b^2 + \overline{b}}{2a} \cdot \left(\frac{b^2}{b^2 + \overline{b}} \cdot \left(\frac{z^{|\beta|}}{b}\right)^2 + \frac{b^2}{b^2 + \overline{b}} \cdot \frac{(z^2)^{|\beta|}}{\overline{b}}\right) = \frac{z^{2|\beta|}}{a} = \frac{z^{|\alpha|}}{a}. \tag{14}$$

□

4.2 Otter Trees

We've already thoroughly discussed the class $\mathcal{B}$ of binary trees. These binary trees are *plane*, in the sense that there the children of an internal node are distinguished: there is a left node and a right node. We now consider the class $\mathcal{V}$ of *Otter tree*, which are binary trees that are *non plane*, using our newly defined $\textsc{MSet}_2$ operator,

$$\mathcal{V} = \mathcal{Z} + \textsc{MSet}_2(\mathcal{V}). \tag{15}$$

The generating function $V(z)$ for Otter trees satisfies the functional equation

$$V(z) = z + \frac{V(z)^2 + V(z^2)}{2},$$

and note that, for this class, we only count external nodes. This combinatorial class does not have a closed form generating function: prior Boltzmann sampler for Otter trees have already informally used approximations [8, §5]; Pivoteau [16] used the fact that $V(z) = 1 - \sqrt{1 - 2z - V(z^2)}$. In practice these approximations yield correct simulations, but theoretically they could introduce a bias. With analytic samplers, this possible bias is corrected by the death rejection; this also gives us more flexibility to choose the approximations.

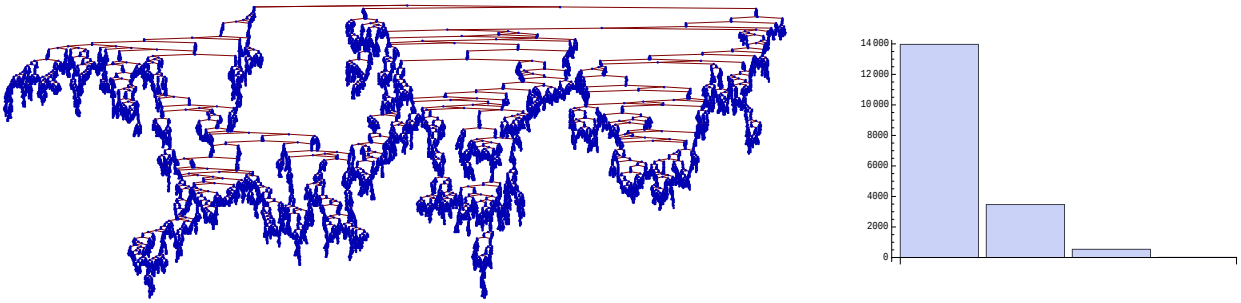

Figure 4. An Otter tree of size $n = 17\,979$ (we were targeting $20\,000 \pm 10\%$), generated in $13s$ on standard laptop computer (Macbook Air 2012). The bar chart summarizes the degree of symmetry of the leaves: the first bar indicates how many leaves are not duplicated; then duplicated once; then four times; then eight times.

Setting up the inequation. For our analytical samplers, we need the values $v_{[i]}$, corresponding to $V(z^{2^i})$, which are defined inductively by the system of inequations

$$\forall i \in \mathbb{N}^+, \quad v_{[i]} \geqslant z^{2^i} + \frac{{v_{[i]}}^2 + v_{[i+1]}}{2}. \tag{16}$$

Because this system is infinite, we are first going to pick a threshold index after which the equations will be approximated; and we will determine a good approximation for the remaining terms.

In order to find solutions, we need an initial interval for z, which need not be especially precise: to this end, $0 \leqslant z \leqslant 1$ suffices (even though it is simple enough to argue that $1/4 < z < 1/2$). The constant part of this recursive inequation is z^{2^i}, thus it makes sense to let $v_{[i]} = Kz^{2^i}$, which we can then inject in our inequation. Dividing both sides by z^{2^i} and factoring, we obtain

$$K \geqslant 1 + \frac{Kz^{2^i}}{2}(K+1). \tag{17}$$

Choosing parameters. At this point we now have two parameters to pick. First we have to find a constant K satisfying Inequation (17); K can be as small as we want, $K > 1$

Once we have picked a threshold i_0, and the constant K which will approximate terms $v_{[i]}$ for $i > i_0$, we can exactly compute the initial terms. This is done by solving exactly the quadratic equations,

$$\frac{1}{2}{v_{[i]}}^2 - v_{[i]} + \left(z^{2^i} + \frac{1}{2}v_{[i+1]}\right) = 0$$

going backwards from $i_0 - 1$ to 0, and with, as we said, the remaining terms $o_{[i]} = Kz^{2^i}$.

The approximations we have taken here will impact the death rate, and we can decrease it by taking any of the following measures: we can pick a higher threshold i_0; we can pick a z that is closer to the singularity; we can use more than the constant part of the equation in the step where we reject $z^{2^{i+1}}$ to approximate the terms beyond the threshold.

This leads to an efficient sampler for Otter trees, of which we have drawn a very large tree in Figure 4. Consider that this allows for interesting empirical analyses of these trees.

5 Conclusion

In this paper, we have proposed to integrate the classical idea of rejection sampling to the Boltzmann sampler model, therefore relaxing the condition that generating functions must be evaluated exactly.

The resulting model, which we call *analytic samplers*, is fully compatible with all prior approaches used in Boltzmann samplers (in particular, these samplers can work well with Pivoteau *et al.*'s oracle), and in fact provides sound theoretical ground by which to allow the routine approximations that have been made in existing Boltzmann samplers.

But beyond that, we also believe the relaxed properties can allow for possible improvements and simplifications in the way the samplers target the size of their output. To illustrate these ideas, we show two types of applications. First, we show for Otter trees, that our samplers allow for much larger approximation to be made with little side-effects. Second, with the example of simply generated trees, we illustrate how tuning can be done in an alternate way, by using the added degree of freedom of exploring points in a region instead of a curve.

The open question is to determine whether these properties can be leveraged for large combinatorial systems: indeed, the initial Boltzmann paper only used as examples combinatorial defined in one or a handful of equations. The real impressive strength of the oracle provided by Pivoteau *et al.* [18] was to be able to handle combinatorial systems with thousands of equations. It remains to be seen if, when navigating wildly more complex polytopes, it is possible to use simple refinements of the ideas we have show for simply generated trees.

Acknowledgments

We would like to acknowledge and thank the considerable feedback we have received from anonymous referees on a prior draft of this work.

References

[1] Frédérique Bassino, Julien David, and Cyril Nicaud. Regal: A library to randomly and exhaustively generate automata. In Jan Holub and Jan Žďárek, editors, *Implementation and Application of Automata*, volume 4783 of *Lecture Notes in Computer Science*, pages 303–305. Springer Berlin Heidelberg, 2007.

[2] Olivier Bodini, Guillaume Moroz, and Hanane Tafat-Bouzid. Infinite Boltzmann samplers and applications to branching processes. In *Proceedings of GASCOM 2012*, pages 1–7, June 2012.

[3] Richard P. Brent. *Algorithms for minimization without derivatives*. Courier Dover Publications, 1973.

[4] Alexis Darrasse. *Structures arborescentes complexes: analyse combinatoire, génération aléatoire et applications*. PhD thesis, Paris 6, 2010.

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

[6] 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.

[7] 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.

[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, 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.

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

[11] 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.

[12] Norman L. Johnson, Adrienne W. Kemp, and Samuel Kotz. *Univariate Discrete Distributions*. Wiley Series in Probability and Statistics. John Wiley & Sons, Inc., 3rd edition, 2005.

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

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

[15] Albert Noack. A Class of Random Variables with Discrete Distributions. *The Annals of Mathematical Statistics*, 21(1):127–132, March 1950.

[16] Carine Pivoteau. Mémoire de master d'informatique. génération aléatoire et modeles de boltzmann. Master's thesis, 2009.

[17] 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.

[18] Carine Pivoteau, Bruno Salvy, and Michèle Soria. Algorithms for combinatorial structures: Well-founded systems and Newton iterations. *Journal of Combinatorial Theory, Series A*, 119(8):1711–1773, 2012.