Efficient sampling of increasing trees

Nadja Azzouz

Institute of Discrete Mathematics and Geometry, TU Wien, Austria

Olivier Bodini

Institut Galilée, Univ. Paris Nord

Francis Durand

Institut Galilée, Univ. Paris Nord

Bernhard Gittenberger

Institute of Discrete Mathematics and Geometry, TU Wien, Austria

Abstract

We present a new exact-size sampler for increasing trees that outputs a tree of size n uniformly at random while avoiding the global coefficient pre-computation required by the classical recursive method of Flajolet *et al.* [17]. The key idea is a hybrid oracle-driven rejection scheme in which local sampling decisions are made using interval bounds on the coefficients, with a fallback to exact recurrence computation only on rare ambiguous events. In the bit-complexity model this yields an expected running time of $O(n \log n)$ and it consumes a number of random bits within $O(n)$ of the Shannon entropy, which is information-theoretically optimal up to lower-order terms. The sampler proceeds in two phases. We first generate the unlabeled rooted ordered shape by recursively sampling node arities and subtree sizes and then draw a uniform permutation of $\{1, \ldots, n\}$ and apply a deterministic increasing-labeling procedure.

2012 ACM Subject Classification Mathematics of computing → Combinatorial algorithms; Mathematics of computing → Enumeration; Mathematics of computing → Generating functions

Keywords and phrases sampling algorithms, bit-complexity, increasing trees, generating functions.

Digital Object Identifier 10.20422/LIPIcs.AofA.2026.8

Funding *Nadja Azzouz*: supported by the Austrian Science Foundation FWF, grant I-6744-N.
Olivier Bodini: supported by the Agence Nationale de Recherche, grant ANR-23-CE48-0014
Francis Durand: supported by the Agence Nationale de Recherche, grant ANR-23-CE48-0014
Bernhard Gittenberger: supported by the Austrian Science Foundation FWF, grant I-6744-N.

1 Introduction

This paper introduces a new algorithm for the exact size uniform random generation of increasingly labeled trees. More precisely, given a combinatorial class of increasing trees, the algorithm samples an increasing tree of size n uniformly at random from all trees of size n. The algorithm avoids the global precomputation of counting sequences required by the classical recursive method [17], while preserving uniformity.

For the analysis of the algorithm, we use the bit-complexity model, which provides a realistic framework for evaluating its efficiency. This model accounts for the cost of random bit generation and arithmetic operations on large integers. We show that the sampler runs in expected time $O(n \log n)$. This matches the information-theoretic lower bound imposed by the entropy of the output distribution, demonstrating that the algorithm is efficient in a strong and practically meaningful sense. This answers a question raised by Luc Devroye (private communication).

We consider a variety $\mathcal{T}$ of increasingly labeled trees specified by a polynomial degree function ϕ. The size of a tree is its number of vertices and the number of trees with n vertices is denoted by t_n. From the seminal paper [3] on increasing trees we know that the

37th International Conference on Probabilistic, Combinatorial and Asymptotic Methods for the Analysis of Algorithms (AofA 2026).
Editor: Konstantinos Panagiotou; Article No. 8; pp. 8:1–8:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

exponential generating function $T_0(z)$ satisfies the differential equation

$$T_0'(z) = \phi(T_0(z)), \qquad T_0(0) = 0, \tag{1}$$

where $\phi(z) = \sum_{k=0}^{d} \phi_k z^k$ is a polynomial with nonnegative coefficients and nonzero constant term and of degree $d \geq 2$.

The classical recursive method of Flajolet, Zimmermann and Van Cutsem [17] provides an exact-size uniform sampler for many labeled classes, including increasing trees. It samples recursively by means of exact counting sequences. However, a direct implementation typically requires precomputing all relevant coefficients up to size n, which is costly for large n (both in arithmetic and memory), and becomes the main bottleneck. Approximate-size approaches like Boltzmann sampling [13, 14, 6] do not directly address the requirement of exact size and uniformity. A related but more specialized result was obtained by Bodini, Durand and Marchal [5], who resolved the exact uniform random generation problem for strict binary increasing trees[1] in $O(n \log(n))$. Their approach relies on a natural bijection between strict binary increasing trees and alternating permutations, which allows for bypassing recursive counting altogether and to design an optimal sampler directly at the permutation level. While extremely efficient in this specific setting, this method is intrinsically tied to the binary case and to the existence of such a bijection. Another specialized approach was presented by Devroye [12] for binary search trees, but under the assumption that basic operations with numbers cost constant time, regardless of the size of the numbers. Moreover, the algorithms were designed for a different purpose and did not output the tree shape.

In this paper, we introduce a hybrid exact sampler that preserves the uniform distribution while avoiding global precomputation in the typical case. At a high level, the sampler proceeds in two phases:

(1) **Shape sampling.** We sample an unlabeled rooted ordered tree shape of size n with the distribution induced by the increasing-tree class, by recursively sampling node arities and subtree sizes.
(2) **Label filling.** We draw a uniform random permutation π of $\{1, \ldots, n\}$ and give a deterministic labelling procedure, which turns the sampled shape into an increasingly labelled tree and yields a labelling that is uniform among all valid increasing labellings of that shape.

The new ingredient lies in the shape phase: instead of using exact coefficients everywhere, we rely on an oracle giving certified bounds for the coefficients and an oracle-driven acceptance rule. When the available precision is insufficient, we fall back to exact coefficient computation via the classical recursive method. We prove that fallbacks are rare enough to preserve optimal expected complexity.

To construct the oracle, we exploit the Puiseux expansion of the solution $T_0(z)$ of (1) around the dominant singularity $z = \rho$. The Puiseux coefficients turn out to be algorithmically computable to any prescribed order and the polynomial order needed for a desired accuracy can be effectively computed up to an exponentially decaying error term.

Outline of the paper. In Section 2 we present the hybrid sampling algorithm for increasing trees, *i.e.*, the algorithms for root-arity sampling, subtree size sampling via a continuous envelope (beta kernel) with acceptance–rejection with oracle-driven Bernoulli sampling, and finally the label-filling step. The subsequent section is devoted to the proof of

[1] Erroneously called strictly increasing binary trees in [5]

correctness and the complexity analysis under the bit complexity model. In Section 4 we briefly introduce theoretical framework for the oracle.

2 The hybrid sampling algorithm

We establish an exact size sampler; i.e. a function $\Gamma_{\mathcal{T}}(n)$ which returns a random uniform increasing tree of size n.

The sampler $\Gamma_{\mathcal{T}}$ retains the general structure of the pure recursive method sampler $\tilde{\Gamma}_{\mathcal{T}}$ [17] but is hybrid in the following sense: in the classical recursive method, every decision (root degree and subtree sizes) is made from exact coefficients, which requires precomputing large tables. Here we instead use an oracle that provides guaranteed bounds on the required coefficients and run an acceptance–rejection scheme that remains exact despite using only bounds. Only when the bounds are too coarse to decide an acceptance step we fall back to the classical recursive computation for that specific coefficient. Thus, the algorithm behaves like an oracle-driven sampler in the typical case, and like the classical recursive sampler only in rare fallback events. Remarkably, $\Gamma_{\mathcal{T}}$ uses the exact recursive sampler $\tilde{\Gamma}_{\mathcal{T}}$ in only a negligible number of cases, achieving an $O(n\log(n))$ running time while guaranteeing an exactly uniform distribution.

In what follows, we will use the following notations: The coefficients of the generating function $T_0(z)$ given by (1) are $t_n = [z^n]T_0(z)$. Similarly, we set $t_n^{(a)} = [z^n]T_0(z)^a$ for any positive integer a. The positive integer m denotes the oracle precision exponent, *i.e.*, the relative error of the output of the oracle is assumed to be of order $O\left(n^{-m}\right)$ with explicit O-constant. Finally, we frequently use the set $\{1, 2, \dots, d\}$ which we denote by $[d]$.

Algorithm 1 Γ: the hybrid random tree sampler

Input: Size n, an oracle for the coefficients $t_n^{(a)}$
Output: $\Gamma(n)$ the shape of a tree of size n sampled uniformly

if $n = 1$ **then**
  return single node tree
end
remaining_size $\leftarrow n - 1$
$a \leftarrow$ SampleArity(remaining_size)
$\mathcal{C} \leftarrow \emptyset$
for $i \leftarrow 1$ **to** a **do**
  if $i < a$ **then**
    $k_i \leftarrow$ NextChildSize($a - i + 1$, remaining_size)
    $\mathcal{C}$.append($\Gamma(k_i)$) `// Recursive call`
    remaining_size $\leftarrow$ remaining_size $- k_i$
  else
    $\mathcal{C}$.append(Γ(remaining_size)) `// Recursive call`
  end
end
return tree with root of arity a and children $\mathcal{C}$

Algorithm 1 delegates all random choices to two subroutines: sampling the node degree via SampleArity and sampling the next child size via NextChildSize. In SampleArity, the candidate is the proposed arity a, and in NextChildSize the candidate is the proposed subtree size k. In both cases, the acceptance probabilities of a candidate have the generic form w/D where w is not computed exactly but only enclosed by certified bounds $w \in [\underline{w}, \overline{w}]$, while the denominator D is a known dominating quantity (with $D \geq \overline{w}$). Algorithm 2 implements this acceptance step by sampling $B \sim \mathrm{Bernoulli}(w/D)$ exactly from the bounds, and only in the rare ambiguous case it falls back to an exact computation of w.

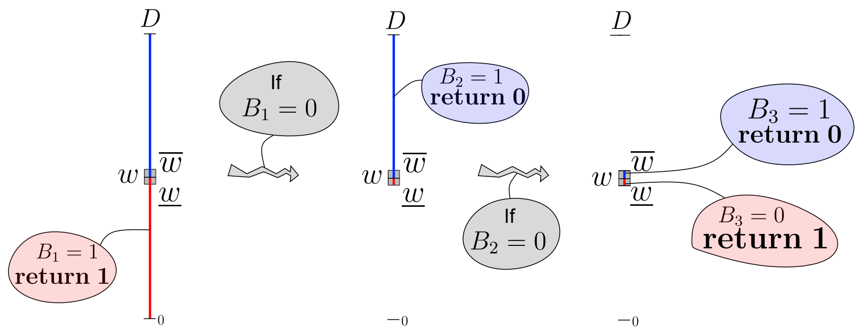

Figure 1 Decision flow for sampling a Bernoulli variable. The red segment indicates immediate acceptance (return 1), the blue one indicates rejection (return 0), and the gray regions represent cases requiring higher precision

Algorithm 2 IntervalBernoulli: exact Bernoulli sampling from certified bounds

Input: a dominating value $D > 0$, bounds $\underline{w}, \overline{w}$ with $0 \leq \underline{w} \leq w \leq \overline{w} \leq D$
Output: $B \sim \mathrm{Bernoulli}(w/D)$

Draw $B_1 \sim \mathrm{Bernoulli}\left(\frac{\underline{w}}{D}\right)$
if $B_1 = 1$ **then**
  return 1
else
  Draw $B_2 \sim \mathrm{Bernoulli}\left(\frac{D - \overline{w}}{D - \underline{w}}\right)$
  if $B_2 = 1$ **then**
    return 0
  else
    `(fallback)` Compute the exact value w
    Draw $B_3 \sim \mathrm{Bernoulli}\left(\frac{w - \underline{w}}{\overline{w} - \underline{w}}\right)$
    if $B_3 = 1$ **then**
      return 1
    else
      return 0
    end
  end
end

We implement all Bernoulli draws by precision-adjusted generation from dyadic interval bounds, see [4, Sec. 4.2].

2.1 Sampling of the node degree

As in the classic recursive method for trees, the arity of each node needs to be determined. From (1) we can extract the recurrence equation $t_{n+1} = \sum_a \phi_a t_n^{(a)}$. That means that the distribution of a – the arity of the root vertex for a tree of size $n+1$ – is proportional to $\phi_a t_n^{(a)}$. Algorithm 3 samples the arity of the root vertex with this distribution.

Algorithm 3 SampleArity: sampling the root degree

Input: target size $n > 1$, degree bound d, interval oracle $\mathcal{O}$
Output: sample $a \in \{1, \ldots, d\} \propto \phi_a t_n^{(a)}$

for $a \in \{1, \ldots, d\}$ **do**
  Query the oracle $\mathcal{O}$ for dyadic bounds $\underline{t_n^{(a)}}, \overline{t_n^{(a)}}$ such that $t_n^{(a)} \in [\,\underline{t_n^{(a)}}, \overline{t_n^{(a)}}]$
Draw A on $[d]$ with $\mathbb{P}[A = a] = \frac{\phi_a \overline{t_n^{(a)}}}{\sum_{a'} \phi_{a'} \overline{t_n^{(a')}}}$.
$B \leftarrow$ IntervalBernoulli $\left(D = \overline{t_n^{(A)}},\ \underline{w} = \underline{t_n^{(A)}},\ \overline{w} = \overline{t_n^{(A)}}\right)$;
if $B = 1$ **then**
  return A
else
  Reject A and resample

Algorithm 3 is an exact accept-reject procedure: A is proposed with weights $\phi_a \overline{t_n^{(a)}}$ and accepted with probability $t_n^{(a)} / \overline{t_n^{(a)}}$, hence the returned arity has distribution $\propto \phi_a t_n^{(a)}$. This coincides with the root–degree distribution of the original recursive method. Moreover, by the oracle precision guarantee, the probability of triggering the exact fallback is at most $1 - \underline{t_n^{(A)}} / \overline{t_n^{(A)}} = O(n^{-m})$. Since d is constant and the proposal weights are dyadic, drawing A on $[d]$ from these weights is implemented by prefix sums and one uniform draw.

2.2 Sampling child subtree sizes

After sampling the root arity a, the next task is to split the remaining size $n - 1$ among the children. This results in the vector of subtree sizes $(k_1, \ldots, k_a)$ with the conditional law induced by the combinatorial specification. We sample the child sizes sequentially. Given b children still to be generated and remaining size s, the size K of the next child has a discrete distribution proportional to

$$f(k) = \frac{t_k t_{s-k}^{(b-1)}}{k!(s-k)!},$$

and $f(k) = 0$ outside the admissible range $\{1, \ldots, s - b + 1\}$. Since computing all f is too costly, we instead perform an accept-reject step using a continuous dominating envelope derived from coefficient asymptotics. The envelope has the shape of the density of a beta distribution. In particular, for bounded degree $b \le d$ and for all admissible sizes, coefficient asymptotics imply an upper bound of the form

$$f(k) \;\le\; C_b\, \rho^{-s} \left(\frac{k}{s}\right)^{\beta-1} \left(1 - \frac{k}{s}\right)^{(b-1)\beta-1}, \qquad 1 \le k \le s - b + 1,$$

where $\beta = 1/(d-1)$ and for a constant $C_b > 0$ depending only on (ϕ, d). This suggests the continuous envelope

$$g(x) \;:=\; C_b\, \rho^{-s} \left(\frac{x}{s}\right)^{\beta-1} \left(1 - \frac{x}{s}\right)^{(b-1)\beta-1}, \qquad x \in (0, s),$$

whose normalized density corresponds to $X/s \sim \mathrm{Beta}(\beta, (b-1)\beta)$. Let $\eta \in [0, s]$ be the turning point of g, given by

$$\eta \;:=\; s\, \frac{\beta - 1}{b\beta - 2},$$

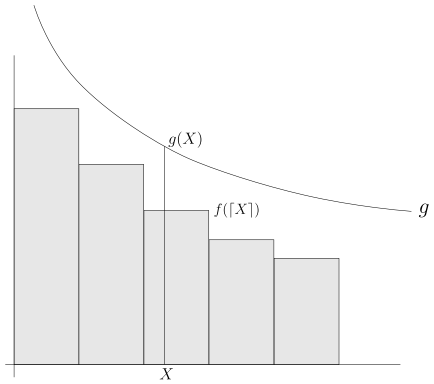

Figure 2 Illustration of the hybrid MC: a point, whose abscissa is X, is drawn uniformly under the graph of g, if it is in a rectangle, that index $\lceil X \rceil$ is returned, otherwise the point is rejected.

and set $\eta := s/2$ in the degenerate case $b\beta = 2$. Since $\beta = \frac{1}{d-1} \leq 1$ and $b \leq d$, the beta kernel is monotone on each side of η, nonincreasing on $(0, \eta]$ and nondecreasing on $[\eta, s)$, where one side may be empty when $\eta \in \{0, s\}$. In the degenerate case, the beta kernel is constant. We then propose an index k from the continuous draw by setting $k = \lceil X \rceil$ when X lies to the left of η (and symmetrically $k = \lfloor X \rfloor$ when it lies to the right). By monotonicity of the beta envelope on each side of η, g dominates the histogram cell associated with the proposed index k. This gives an acceptance probability of the proposed k of

$$\alpha(k, X) = \frac{f(k)}{g(X)} \leq 1.$$

The resulting procedure avoids evaluating the full discrete distribution: it only requires the single value $f(k)$ at the proposed index and the envelope value $g(X)$, together with Bernoulli sampling from the ratio $\alpha(k, X)$.

▶ **Assumption 1.** *We assume access to an exact sampler for $X \sim \mathrm{Beta}(\beta, (b-1)\beta)$ that supports precision-adjusted generation in the following sense. For any target precision $p = O(\log n)$, it returns in time $O(\log n)$ a dyadic window $I = [\underline{X}, \overline{X}]$ containing X whose endpoints are adjacent p-bit dyadic floating-point numbers. Moreover, the window is distributed according to the target density $q(x) \propto g(x)$, namely*

$$\mathbb{P}(\ I = [\underline{X}, \overline{X}]\ ) = \frac{\int_{\underline{X}}^{\overline{X}} g(x)\, dx}{\int_0^s g(x)\, dx}.$$

Such samplers can be obtained by combining arbitrary-precision sampling principles with classical beta variate generation; see [11, 10].

To implement the acceptance step with dyadic arithmetic, we work with the window $I = [\underline{X}, \overline{X}]$ returned by the precision-adjusted beta sampler and reject windows that cross η. Since the endpoints are adjacent dyadic floats, I contains no integer and therefore the proposed index k defined by $\lceil X \rceil$ (or $\lfloor X \rfloor$) is constant on I.

For the remaining events, the domination property holds pointwise on I, so $g(x) \geq f(k)$ for all $x \in I$. Define

$$D := \frac{1}{\overline{X} - \underline{X}} \int_{\underline{X}}^{\overline{X}} g(x)\, dx.$$

Then $D \geq f(k)$, and therefore the call IntervalBernoulli$(D, \underline{f(k)}, \overline{f(k)})$ is well-defined, as we have the oracle bounds $\underline{f(k)} \leq f(k) \leq \overline{f(k)}$.

Algorithm 4 NextChildSize

Input: remaining children b, remaining size $s \geq b$, degree d, interval oracle $\mathcal{O}$
Output: integer size k of the next child

```
Draw U ~ Beta(β, (b − 1)β)
X ← sU and obtain a dyadic window [X̲, X̄] containing X
// Split at η, where g switches monotonicity
if X̄ ≤ ⌊η⌋ then
│ k ← ⌈X⌉
else if X̲ ≥ ⌈η⌉ then
│ k ← ⌊X⌋
else
│ // window straddles η
│ Reject X and resample
D ← (∫_{X̲}^{X̄} g(x) dx) / (X̄ − X̲)
Query O for dyadic bounds f̲(k), f̄(k)
B ← IntervalBernoulli(D, w̲ = f̲(k), w̄ = f̄(k))
if B = 1 then
│ return k
else
│ Reject X and resample
```

Let τ be the rooted ordered tree shape produced by Algorithm 1 on input n. We construct a labelling ℓ on the vertex set, that is uniform among all increasing labellings of τ, as follows. First compute all subtree sizes $|\tau_v|$ by one postorder traversal. Finally, we draw a uniform random permutation π of $[n]$, view it as an ordered list of distinct labels, and call $\mathrm{Label}(\mathrm{root}, \pi)$, where $\mathrm{Label}(v, L)$ is defined recursively by:
(i) set $\ell(v) \leftarrow \min(L)$,
(ii) let L' be L with this minimum removed,
(iii) if $v_1, \ldots, v_k$ are the children of v in the given order and $s_i := |\tau_{v_i}|$, split L' into consecutive blocks $(L_1, \ldots, L_k)$ with $|L_i| = s_i$ and recurse with $\mathrm{Label}(v_i, L_i)$ for all i.
Since π is uniform on S_n and this procedure maps exactly the same number of permutations π to each increasing labelling of τ (namely $\prod_v |\tau_v|$), the resulting labelling is uniform over all increasing labellings of τ.

3 Complexity analysis (bit–complexity model)

We analyze Algorithm 1 in the (randomized) *bit–complexity model* [2, 7]. We report in this section the expected number of elementary bit operations as well as the expected number of unbiased random bits consumed. Consuming one unbiased random bit requires reading one bit from the random source, hence

$$\#\text{random bits consumed} \;\leq\; \#\text{bit operations}. \tag{2}$$

Throughout this section, the degree bound d and the oracle precision exponent $m \geq 5$ are fixed constants. All hidden constants may depend on (ϕ, d, m) but not on n. Furthermore, log denotes the base-2 logarithm.

▶ **Theorem 1.** *Algorithm 1 returns an increasing tree of size n with the uniform distribution on $\mathcal{T}_n$ using*

$$O(n \log n) \text{ bit operations} \qquad \text{and} \qquad n \log n + O(n) \text{ random bits}$$

in expectation. Thus the random bit complexity of the sampler matches the information-theoretic lower bound up to an additive $O(n)$ term, and its expected bit-operation cost is optimal up to a constant factor.

Proof. Let t_n be the number of increasing trees of size n. By the known coefficient asymptotics for increasing trees [3] and Stirling's formula,

$$\log t_n = \log(n!) + n\log(1/\rho) + (\beta - 1)\log n + O(1) = n\log n + n\big(\log(1/\rho) - \log e\big) + O(\log n).$$

Any exact uniform sampler must use at least the Shannon entropy $\log t_n$ random bits in expectation. In particular, $\log t_n = n\log n + O(n)$. So, once we have shown for our Algorithm to perform within this consumption, we have justified the optimality claim.

- *Dyadic floating-point numbers and adaptive precision.* All inexact quantities manipulated by the sampler (oracle bounds, acceptance thresholds, proposal windows, etc.) are represented as signed dyadic floating-point numbers. Crucially, the sampler only compares *ratios* of coefficients at a fixed size, so any common factor such as ρ^{-n} cancels and is never explicitly formed. In addition, the working precision is chosen *locally* with the current subproblem size: when the sampler is called on a subtree of size s, all oracle queries and interval computations in that call are performed with mantissa length

  $$p(s) := \lceil c_m \log s \rceil = \Theta(\log s),$$

  where $c_m > 0$ is chosen so that certified relative widths of order $O(s^{-m})$ are maintained under a constant number of arithmetic operations using directed rounding.
- *Bernoulli sampling from bounds.* Algorithm 2 samples $B \sim \text{Bernoulli}(p^\star)$ from dyadic bounds $\underline{p} \le p^\star \le \overline{p}$ and triggers the exact fallback only when the uniform draw falls in the residual region of mass at most $\overline{p} - \underline{p}$. In our setting, directed interval arithmetic yields $\overline{p} - \underline{p} = O(s^{-m})$ at size s, so the fallback probability is $O(s^{-m})$; see [4, Sec. 4.2] and [11, 10].
- *Bit complexity of arithmetic.* Let $M_\times(p)$ denote the bit complexity of multiplying two p-bit integers. (Addition/subtraction costs $O(p)$.) A constant number of floating-point operations on p-bit mantissas (add, multiply, divide, comparison, normalization, and directed rounding) costs

  $$O\big(M_\times(p)\big) \qquad \text{bit operations.}$$

  With schoolbook multiplication $M_\times(p) = O(p^2)$, hence

  $$M_\times(p(s)) = O(\log^2 s).$$

- *Oracle queries.* By construction of the oracle in Section 4, one query at size s returns dyadic bounds with relative width $O(s^{-m})$ using $O(M_\times(p(s)))$ bit operations.
- *Fallbacks.* Each invocation of Algorithm 2 triggers the exact fallback with probability $O(s^{-m})$ at subproblem size s. If fallback occurs, the exact values needed for that decision

can be computed in $O(s^4 \log^2 s)$ bit operations by the analysis of the classical recursive method in [9]. Hence the expected fallback overhead of one invocation is

$$O(s^{-m}) \cdot O(s^4 \log^2 s) = O(s^{4-m} \log^2 s).$$

For $m \geq 5$, this is $O\left(\frac{\log^2 s}{s}\right)$ and therefore gets absorbed by the $O(\log^2 s)$ bit cost of the interval arithmetic performed in the call.

- *Toll per call.* Let $\mathrm{toll}(s)$ denote the expected work performed at the root of a call on size s (i.e., sampling the arity and sampling the child sizes, excluding recursive calls to smaller subtrees). Since d is constant, SampleArity performs $O(1)$ oracle queries and $O(1)$ interval operations at precision $p(s)$. Similarly, each call to NextChildSize performs $O(1)$ oracle queries and $O(1)$ interval operations at precision $p(s)$, and it calls IntervalBernoulli a constant number of times with $p(s)$-bit dyadic parameters; the expected number of proposals is $O(1)$ (by the constant acceptance probability established in the analysis of the envelope). Therefore,

$$\mathrm{toll}(s) = O\big(M_\times(p(s))\big) = O(\log^2 s).$$

- *Label filling cost.* Generating a uniform random permutation of $\{1, \ldots, n\}$ can be done with $n \log n + O(n)$ random bits and $O(n \log n)$ bit operations (e.g. MergeShuffle [1]). Given the sampled shape, we first compute all subtree sizes by one postorder traversal. The permutation-driven recursive split then assigns labels using $O(1)$ work per vertex on $O(\log n)$-bit indices, hence it costs $O(n \log n)$ bit operations overall.

Let F_n denote the expected number of bit operations needed to sample the shape of a uniform increasing tree of size n (Algorithm 1 without the final permutation/label-filling step). Then F_s satisfies a recurrence of the form

$$F_s \;=\; \mathrm{toll}(s) \;+\; \sum_{k=1}^{s-1} \omega_{s,k}\, F_k, \tag{3}$$

where $\omega_{s,k}$ is the expected number of recursive calls on size k issued by a call on size s. Let A be the arity of the root and K the size of a (distinguished) child subtree,

$$\omega_{s,k} \;=\; \mathbb{E}\big[\#\{\text{children of size } k\}\big] \;=\; \sum_{a \in [d]} \mathbb{P}(A = a)\, a\, \mathbb{P}(K = k \mid A = a).$$

Using the standard recursive decomposition for increasing trees, this yields

$$\omega_{s,k} = \sum_{a \in [d]} a\, \frac{\phi_a\, t_{s-1}^{(a)}}{t_s} \binom{s-1}{k} \frac{t_k\, t_{s-1-k}^{(a-1)}}{t_{s-1}^{(a)}} = \binom{s-1}{k} \frac{t_k}{t_s} \sum_{a \in [d]} a \phi_a\, t_{s-1-k}^{(a-1)}. \tag{4}$$

In our setting, $(\omega_{s,k})$ admits a shape function $\omega(x)$, in the sense of Roura's continuous recursive definitions [20, Def. 3.1.], given by

$$\omega(x) = \frac{d}{d-1}\, x^{\frac{1}{d-1}-1}, \qquad x \in (0,1),$$

and the recurrence (3) falls under the scope of Roura's Continuous Master Theorem [20, Thm. 3.3, case (3)]. With $\mathrm{toll}(s) = O(\log^2 s)$ ($a = 0$, $c = 2$ in Roura's notation) and

$$\varphi(x) = \int_0^1 \omega(z)\, z^x\, dz = \frac{d}{(d-1)x+1},$$

we have $1-\varphi(0) = 1-d < 0$, and the unique solution of $\varphi(\alpha) = 1$ is $\alpha = 1$. Hence,

$$F_n = O(n). \tag{5}$$

By (2), the number of random bits used to sample the shape is also $O(n)$ in expectation.

Combining $F_n = O(n)$ for the shape phase with the label-filling cost yields $O(n \log n)$ bit operations in expectation. The random bits are $O(n)$ for shape sampling plus $n \log n + O(n)$ for the permutation, hence $n \log n + O(n)$ in expectation. ◀

4 The oracle for coefficient estimates

4.1 Definition and theoretical framework

In this section we will present the oracle that is used in the sampling algorithms in Section 2. We start with the definition.

▶ **Definition 2** (Coefficient approximation oracle). *As in Section 2, let $t_n^{(a)} = [z^n]T_0(z)^a$, where $T_0(z)$ is given by (1). A* coefficient approximation oracle *is a procedure $\mathcal{O}$ that takes as input integers $n, m \geq 1$ and an integer $a \in \{1, \dots, d\}$ and returns two nonnegative rational numbers (technically these are dyadic numbers) $\underline{t_n^{(a)}}$ and $\overline{t_n^{(a)}}$ such that*

$$\underline{t_n^{(a)}} \leq t_n^{(a)} \leq \overline{t_n^{(a)}},$$

and the relative length of the bounding interval is polynomially small, i.e., there exists an effectively computable constant $C > 0$, which is independent of n and a, such that

$$\frac{\overline{\tau_n^{(a)}} - \underline{\tau_n^{(a)}}}{\underline{\tau_n^{(a)}}} \leq \frac{C}{n^m}. \tag{6}$$

Equivalently, we may define $\mathcal{O}$ to output bounds for the counting numbers $t_n^{(a)} = n!\tau_n^{(a)}$; the relative precision is unchanged.

We will construct the oracle upon the Puiseux expansion of $T(z)$ at its dominant singularity ρ. But we have to require a certain condition to make it work. Before we start, we recall the singular behaviour of $T_0(z)$. The first part of the following theorem is a well-known result from [3].

▶ **Theorem 3.** *Assume that $\phi(z)$ is a polynomial of degree $d \geq 2$ with $\phi_d > 0$ and $\phi_0 > 0$. Let $T_0(z)$ denote the solution of (1) defined locally near 0 and $\rho > 0$ its dominant singularity, then*

$$T_0(z) \sim ((d-1)\phi_d(\rho - z))^{-1/(d-1)}.$$

In particular, ρ is a pole if $d = 2$ and an algebraic branch point of order $d-1$ otherwise.

If $T(z)$ is an analytic continuation of $T_0(z)$ and $z_0 \in \mathbb{C}$ one of its singular points, then the analogous formula holds (with ρ replaced by z_0).

Proof. As $\phi(z)$ is Lipschitz continuous near every $z \in \mathbb{C}$, the existence and uniqueness theorem of Picard-Lindelöf implies that $T(z)$ can only be singular at some z_0 if $\lim_{z \to z_0} T(z) = \infty$. But then we have

$$\frac{1}{(d-1)T(z)^{d-1}} = \int_z^{z_0} \frac{T'(t)}{T(t)^d}\, \mathrm{d}t \sim \int_z^{z_0} \phi_d \, \mathrm{d}t = \phi_d(z_0 - z)$$

which implies the assertion. ◀

Under a convergence condition on the Puiseux series of $T_0(z)$ around ρ, we can exploit the coefficients of the Puiseux expansion for our oracle, as the next two Propositions show.

▶ **Proposition 4.** *Let*

$$\sum_{j\geq 0} c_j \left(1-\frac{z}{\rho}\right)^{(j-1)/(d-1)}$$

be the Puiseux expansion of $T_0(z)$ around its dominant singularity ρ. Moreover, assume that the function

$$G(u) = \sum_{j\geq 0} c_j u^j \tag{7}$$

is analytic at 0 and the series has radius of convergence $R_G > 1$. Then, for every $n \geq 0$ we have

$$[z^n]T_0(z) = \rho^{-n} \sum_{j\geq 0} c_j \binom{\frac{j-1}{d-1}}{n} (-1)^n, \quad \textit{where } \binom{\alpha}{n} := \frac{\alpha(\alpha-1)\cdots(\alpha-n+1)}{n!}.$$

Proof. We are here in the setting of Weierstraß's double-series theorem, see [18, p. 83, Theorem 3]. Indeed, we have a series

$$\tilde{T}(z) = T_0(z) - c_0\left(1-\frac{z}{\rho}\right)^{-1/(d-1)} = \frac{G(u)-c_0}{u} = \sum_{j\geq 1} f_j(z)$$

with $f_j(z) = c_j \left(1-\frac{z}{\rho}\right)^{(j-1)/(d-1)}$. The functions $f_j(z)$ are binomial series and thus converge (uniformly and absolutely) for $|z| < \rho$. If the series $\tilde{T}(z)$ converges uniformly in a domain of the form $|z| \leq \rho_1 < \rho$, then Weierstraß's double-series theorem implies the claim.

As $f_j(z) = c_j u^{j-1}$ and $R_G > 1$, the series $\tilde{T}(z)$ converges uniformly for

$$|u| = \left|1-\frac{z}{\rho}\right|^{1/(d-1)} \leq 1+\varepsilon,$$

provided that ε is sufficiently small. So, we indeed have uniform convergence of $\tilde{T}(z)$ for $|z| \leq \varepsilon\rho < \rho$. ◀

This fact shows that we can express the coefficient $[z^n]T_0(z)$ as a sum using only the Puiseux coefficients of the expansion at ρ. This insinuates that we can use the truncated sum to approximate $[z^n]T_0(z)$ to arbitrary presision. The next proposition shows that we can indeed do that effectively.

▶ **Proposition 5.** *Let $G(u)$ be the function given in (7) and assume it has radius of convergence $R_G > 1$. Fix an integer $K \geq 0$ and let J be the smallest integer such that*

$$J \geq \left\lceil \frac{K+1}{\beta} \right\rceil \qquad \textit{and} \qquad \frac{J}{d-1} \notin \mathbb{N}.$$

Set

$$S_J(n) := \rho^{-n} \sum_{j=0}^{J} c_j \binom{\frac{j-1}{d-1}}{n} (-1)^n.$$

Then, there exist constants C_K and D_K such that for sufficiently small $\eta > 0$ the inequality

$$|[z^n]T_0(z) - S_J(n)| \le C_K \rho^{-n} n^{-K-1} + D_K (1+\eta)^{-n}\rho^{-n}$$

holds for all $n > 15$. The inequality is true with

$$C_K = \frac{2M}{r^{J+1}}\left(1 + 2^{2+\frac{J}{d-1}} \cdot \frac{\Gamma\left(\frac{J}{d-1}+1\right)}{\pi}\right)$$

where $M := \sup_{|u|=r} |G(u)|$ and some $0 < r < R_G$.

Under the assumption $R_G > 2^{1/(d-1)}$ the assertion is true for C_K as given above with $r = 2$, $D_K = 2Mr^{-J-1}(2+\eta)^{J/(d-1)}$ and any η satisfying $0 < \eta < R_G - 2^{1/(d-1)}$.

Proof. First, note that $u = (1 - z/\rho)^{1/(d-1)}$. By our assumption that $G(u)$ is analytic in $|u| < R_G$ and $R_G > 1$, the Puiseux series of $T_0(z)$ converges in the domain

$$D := \left\{z \in \mathbb{C} \mid |1 - z/\rho| < (R_G - \varepsilon)^{d-1} \text{ and } z - \rho \notin \mathbb{R}\right\}$$

for some sufficiently small $\varepsilon > 0$. In this domain we apparently have $|u| < R_G - \varepsilon$. This implies $T_0(z) = (1 - z/\rho)^{-1/(d-1)} G(1 - z/\rho)$ for $z \in D$. Now, let $r := R_G - \varepsilon$ and $M := \sup_{|u|=r} |G(u)|$. Then, by Cauchy's estimate, we get $|c_j| \le Mr^{-j}$. Next, consider $D' := \{z \in D \mid |u| \le r/2\}$. For $z \in D'$ (resp. u in the disk with radius $r/2$) we obtain for any $L > 0$ the bound

$$\left|\sum_{j>L} c_j u^j\right| \le \sum_{j>L} Mr^{-j}|u|^j \le 2Mr^{-(L+1)}|u|^{L+1}. \tag{8}$$

Consider the function

$$R(z) = T_0(z) - P(z) = T(z) - \sum_{j=0}^{J} c_j \left(1 - \frac{z}{\rho}\right)^{(j-1)/(d-1)} = \sum_{j>J} c_j \left(1 - \frac{z}{\rho}\right)^{(j-1)/(d-1)},$$

which is well-defined if z is sufficiently close to ρ, and apply (8) to it. This yields

$$|R(z)| \le 2Mr^{-(J+1)} \left|1 - \frac{z}{\rho}\right|^{J/(d-1)}, \qquad \text{for } z \in D'. \tag{9}$$

Next, we follow the proof of the big-O transfer theorem (see [15]). Henceforth, choose the classical contour (keyhole contour complemented by a circular arc, see [16, p. 390]) such that the boundary points of the major circular arc still lie in D'. The contour consists of the inner arc γ_1 of radius $1/n$ and centered at ρ, the rectilinear parts, γ_2 and its complex conjugate, and the outer arc γ_3 of radius $\rho(1+\eta)$, with η sufficiently small, and centered at 0. We use

$$[z^n]T_0(z) = [z^n]P(z) + \frac{1}{2\pi i}\int_{\gamma_2 \cup \gamma_1 \cup \overline{\gamma_2}} \frac{R(v)}{v^{n+1}}\,\mathrm{d}v + \frac{1}{2\pi i}\int_{\gamma_3} \frac{T_0(v) - P(v)}{v^{n+1}}\,\mathrm{d}v. \tag{10}$$

As both the inner arc and the rectilinear parts lie entirely in D', the bound (9) is applicable. For the inner arc γ_1 this gives

$$\frac{1}{2\pi}\left|\int_{\gamma_1} \frac{R(v)}{v^{n+1}}\,\mathrm{d}v\right| \le \frac{2Mr^{-J-1}}{\left(1 - \frac{1}{n}\right)^{n+1}} n^{-\frac{J}{d-1}-1} \le 2Mr^{-J-1} n^{-\frac{J}{d-1}-1} \rho^{-n}$$

where the last inequality holds for $n > 15$. For the rectilinear parts, we have $v = 1 + te^{i\phi}/n$, with the chosen $0 < \phi < \pi/2$ of the keyhole contour, and where the integration goes from $t = 1$ to $t = En$ such that $1 + tE$ is a boundary point of the outer arc.

$$\frac{1}{2\pi}\left|\int_{\gamma_2\cup\overline{\gamma_2}} \frac{R(v)}{v^{n+1}}\,\mathrm{d}v\right| \leq \frac{2M}{\pi r^{J+1}\rho^n} n^{-\frac{J}{d-1}-1}\int_1^\infty w^{\frac{J}{d-1}}\left|1+\frac{we^{i\phi}}{n}\right|^{-n-1}\mathrm{d}w$$
$$\leq \frac{2M}{\pi r^{J+1}\rho^n} n^{-\frac{J}{d-1}-1}\int_0^\infty w^{\frac{J}{d-1}}e^{-\frac{w}{2}}\,\mathrm{d}w$$
$$\leq \frac{2^{J/(d-1)+2}M\Gamma\left(\frac{J}{d-1}+1\right)}{\pi r^{J+1}\rho^n} n^{-\frac{J}{d-1}-1}$$

Altogether, we obtain herewith the desired bound with

$$C_K = \frac{2M}{r^{J+1}}\left(1+\frac{2^{\frac{J}{d-1}+2}\Gamma\left(\frac{J}{d-1}+1\right)}{\pi}\right)$$

as claimed.

Let us turn to the outer arc γ_3. Here $|z| = \rho(1+\eta)$ and $\arg(z) \geq \varphi$ for some fixed $\varphi > 0$. $R(z)$ does not converge for all $z \in \gamma_3$ except if the circle $|z| = \rho$ lies entirely inside the convergence domain of $G(u)$. This is equivalent to $R_G > 2^{1/(d-1)}$. In this case we can rely on (9) for the outer arc as well and get the claimed upper bound.

In the general case we must use a generic bound for the analytic continuation $T(z)$ of $T_0(z)$ instead of (9). As γ_3 is compact and lies in the analyticity domain of $T(z)$, there is a constant C such that $|T(z)| \leq C$ for $z \in \gamma_3$. Moreover, the length of the integration contour is 2π, which cancels out with the prefactor of the Cauchy integral. Thus, we obtain

$$\frac{1}{2\pi}\left|\int_{\gamma_3} \frac{T(v)}{v^{n+1}}\,\mathrm{d}v\right| \leq \frac{C}{(1+\eta)^n\rho^n}.$$

A similar computation for $P(z)$ yields

$$\frac{1}{2\pi}\left|\int_{\gamma_3} \frac{P(v)}{v^{n+1}}\,\mathrm{d}v\right| \leq \frac{\left(\frac{2+\eta}{r}\right)^{\frac{J}{d-1}+1}}{\left(\frac{2+\eta}{r}-1\right)(1+\eta)^n\rho^n}.$$

Finally, the missing coefficient in (10) can be evaluated exactly as $[z^n]P(z) = S_J(n)$. As we know the behaviour of $T_0(z)$ near ρ, we infer that $\tau_n^{(a)}$ satisfies an asymptotic relation of the form $\tau_n^{(a)} = [z^n](T_0(z))^a \asymp \rho^{-n}n^{a/(d-1)-1}$. Choosing K large enough and then M accordingly, we can achieve a relative error of desired order by a similar computation. ◀

We close this section with a few remarks: Proposition 5 is the base of the oracle. The constant K governs the desired accuracy and from this we derive the explicitly computable constants C_K and D_K that bound the error term of the coefficient approximation. It only depends on the chosen accuracy parameter J and the numerically approximable constant M. The drawback of this proposition is that so far we have no control on the constants D_K and η except under the condition $R_G > \rho + \rho^2$. The reason is that η depends on the location of the subordinate singularities of $T(z)$ and C on a general bound for $T(z)$. It is reasonable to believe that this bound is closely related to the local bound of the analytic continuation $T(z)$ near ρ that is given in Theorem 3 and which would give $((d-1)\phi_d\eta)^{-1/(d-1)}$. Moreover, the unknown term decays exponentially such that we can expect that for practical purposes doubling C_K will do the job, even if $R_G \leq \rho + \rho^2$.

4.2 Applying the oracle – an example

Consider the class of increasing trees with degree polynomial $\phi(t) = (t+1)^2(t^2+2)$. We can compute ρ via the functional inverse

$$F(w) = \int_0^w \frac{1}{\phi(v)}\,\mathrm{d}v$$

of $T_0(z)$ by

$$\rho = \lim_{w\to\infty} F(w) = \frac{1}{3} + \frac{1}{9}\log 2 - \frac{\pi\sqrt{2}}{36}.$$

Moreover, after the variable transformation from z to u, the Puiseux expansion at ρ becomes $G(u) = uT(\rho(1-u^3))$ and the differential equation (1) translates to

$$D(G(u),u) := uG'(u) - G(u) - (d-1)\rho u^d \phi\left(\frac{G(u)}{u}\right) = 0,$$

where in our case $d = 4$.

We are first interested in the coefficients of the Puiseux expansion of $T_0(z)$ at ρ. To get them, note that by Theorem 3 we must have $c_0 = ((d-1)\rho\phi_d)^{-1/(d-1)} = (3\rho)^{-1/3}$. Next, consider the function

$$G_n(u) = \sum_{j=0}^{n-1} c_j u^j + xu^n.$$

An easy calculation shows

$$[u^n]G_n(u) = \left(n + \frac{d}{\phi_d} - 1\right)x + \eta_n = (n+3)x + \eta_n$$

where η_n depends only on $c_0, \ldots, c_{n-1}$. This gives the recursion $c_n = -\eta_n/(n+3)$ for the desired coefficients.

Note that $F(w)$ can be analytically continued to complex w and has singularities exactly at the zeros $a_1, \ldots, a_d$ of $\phi(t)$. These singularities are either logarithmic or a sum of a logarithmic and polar singularity. In particular, this means that $F(w)$ is multivalued, like $T(z)$. Hence, the actual value of $F(T(z))$ depends on the chosen branches and equals $z + \lambda$, where $\lambda \in \Lambda = \{a\omega_1 + b\omega_2 + c\omega_3 \mid a,b,c \in \mathbb{Z}\}$ with $\omega_j/(2\pi i)$ being the residues of $1/\phi(t)$ at the double pole at $a_1 = -1$ and the simple poles at $a_2 = i\sqrt{2}$ and $a_3 = -i\sqrt{2}$. An easy calculation yields

$$\omega_1 = \frac{4\pi i}{9}, \qquad \omega_2 = -\frac{\pi\sqrt{2}}{18} - \frac{2\pi i}{9}, \qquad \omega_3 = \frac{\pi\sqrt{2}}{18} - \frac{2\pi i}{9}.$$

As the only possible singularities of $T(z)$ can be at $\rho + \lambda$ with $\lambda \in \Lambda$, there is no singularity closer to ρ than $\rho + \lambda^*$ where λ^* is the element of minimal modulus in Λ, which is $\lambda^* = \pi\sqrt{2}/9 = \omega_3 - \omega_2$. Thus

$$R_G \geq \left|1 - \frac{\rho + \lambda^*}{\rho}\right|^{1/3} = \left(\frac{\lambda^*}{\rho}\right)^{1/3} \approx 1.19824503.$$

As $R_G > 1$ Proposition 5 is applicable. However, we have $R_G \leq 2^{1/3} \approx 1.25992$ and so we do not have explicit multiplicative constants for the exponentially decaying error term.

5 Conclusion

In this work, we presented an efficient sampling algorithm that is of hybrid nature, defaulting to the exact computation of the coefficients only in rare cases, and otherwise queries an oracle for coefficient estimates of adjustable precision. The use of the oracle avoids expensive pre-computations. In the bit-complexity model, the sampler runs in expected $O(n \log n)$ bit operations. And its expected random-bit consumption is $n \log n + O(n)$, matching the Shannon-entropy lower bound up to lower order terms.

The oracle uses the singular structure of the generating function $T_0(z)$ of the model. In Theorem 3, we may take $T(z)$ to be the complete analytic continuation of $T_0(z)$ (in the usual sense of analytic continuation along paths, see [8, Ch. IX, Def. 2.7]), which is defined on the maximal Riemann surface $\mathcal{S}$ such that (i) $T(z)$ is analytic on $\mathcal{S}$ and (ii) for all z in the analyticity domain of the power series $T_0(z)$ we have $T_0(z) = T(z)$. In view of this, Theorem 3 has some implications on the geometry of $T(z)$ near ρ and the distribution of singularities: At ρ the Riemann surface branches into $d-1$ sheets and the Puiseux expansion can be seen on a power series in $u = (1 - z/\rho)^{1/(d-1)}$ where the transformation from z to u maps the $d-1$ sheets into a single one. It can be shown that the singularities of $T(z)$ on the principal sheet form a lattice and so, the $d-1$ sheets altogether yield a discrete set of singularities. The one closest to ρ determines the domain of convergence of the Puiseux series and it can be shown that there is only a finite number of points to check in order to find out whether this domain is large enough for the oracle to work in the sense that the constants involved in the error terms are effectively computable. So far, we have no characterization of the polynomials $\phi(t)$ for which the domain of convergence of the Puiseux series is sufficiently large to guarantee effectively computable estimates for the exponential error term. But if the domain is too small, it affects only the exponentially small error term such that we can expect the output to still be useful in practice.

We expect that the algorithms are of much wider applicability. An extension to more general varieties of increasing trees, where $\phi(t)$ an entire function should be rather straightforward. This would at least cover classical varieties like recursive trees. In a similar way, we expect that the sampling algorithm (and to some extent the oracle) can be adapted to other frameworks, like certain classes of simply generated trees, or even classes with analytic perturbations, like Otter trees [19]. All this is ongoing research.

References

1. Axel Bacher, Olivier Bodini, Alexandros Hollender, and Jérémie Lumbroso. Mergeshuffle: a very fast, parallel random permutation algorithm. *arXiv preprint arXiv:1508.03167*, 2015.
2. Karim Belabas. Advanced computational number theory. Course Notes, University of Bordeaux, 2013. Chapter 1 covers Bit-Complexity and Randomized Algorithms. URL: `https://www.math.u-bordeaux.fr/~kbelabas/teach/N1MA9W11/book.pdf`.
3. François Bergeron, Philippe Flajolet, and Bruno Salvy. Varieties of increasing trees. In *Proceedings of CAAP'92 (Colloquium on Trees in Algebra and Programming)*, volume 581 of *Lecture Notes in Computer Science*, pages 24–48. Springer, 1992.
4. Olivier Bodini and Francis Durand. Optimal random bit complexity in efficient sampling of set partition-like structures. In *International Workshop on Combinatorial Algorithms*, pages 405–417. Springer, 2025.
5. Olivier Bodini, Francis Durand, and Philippe Marchal. Optimal generation of strictly increasing binary trees and beyond, 2024. Proceedings of GASCom 2024. URL: `https://arxiv.org/abs/2406.16396`, `arXiv:2406.16396`.

6 Olivier Bodini, Olivier Roussel, and Michèle Soria. Boltzmann samplers for first-order differential specifications. *Discrete Applied Mathematics*, 160(18):2563–2572, 2012. doi:10.1016/j.dam.2012.05.022.

7 Richard P. Brent and Paul Zimmermann. *Modern Computer Arithmetic*, volume 18 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, Cambridge, UK, 2010. doi:10.1017/CBO9780511921698.

8 John B. Conway. *Functions of one complex variable*, volume 11 of *Graduate Texts in Mathematics*. Springer-Verlag, New York-Berlin, second edition, 1978.

9 Alain Denise and Paul Zimmermann. Uniform random generation of decomposable structures using floating-point arithmetic. volume 218, pages 233–248. 1999. Caen '97. URL: https://doi.org/10.1016/S0304-3975(98)00323-5.

10 Luc Devroye. Nonuniform random variate generation. In Shane G. Henderson and Barry L. Nelson, editors, *Handbooks in Operations Research and Management Science: Simulation*, volume 13, pages 83–121. Elsevier, 2006. doi:10.1016/S0927-0507(06)13005-4.

11 Luc Devroye and Claude Gravel. Random variate generation using only finitely many unbiased, independently and identically distributed random bits. *CoRR*, 2020. URL: https://arxiv.org/abs/1502.02539, arXiv:1502.02539.

12 Luc Devroye and John Michael Robson. On the generation of random binary search trees. *SIAM J. Comput.*, 24(6):1141–1156, 1995. doi:10.1137/S0097539792224954.

13 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. doi:10.1017/S0963548304006315.

14 Philippe Flajolet, Éric Fusy, and Carine Pivoteau. Boltzmann sampling of unlabelled structures. In *Proceedings of ANALCO'07*, pages 201–211. Society for Industrial and Applied Mathematics, 2007. Also available as a section in Analytic Combinatorics texts.

15 Philippe Flajolet and Andrew Odlyzko. Singularity analysis of generating functions. *SIAM J. Discrete Math.*, 3(2):216–240, 1990. doi:10.1137/0403019.

16 Philippe Flajolet and Robert Sedgewick. *Analytic Combinatorics*. Cambridge University Press, 2009.

17 Philippe Flajolet, Paul Zimmermann, and Bernard Van Cutsem. A calculus for the random generation of labelled combinatorial structures. *Theoretical Computer Science*, 132, 1994. doi:10.1016/0304-3975(94)90226-7.

18 Konrad Knopp. *Theory of Functions, Parts I and II: Two Volumes Bound as One*. Dover Publications, New York, 1996. Translated by Frederick Bagemihl.

19 Richard Otter. The number of trees. *Ann. of Math. (2)*, 49:583–599, 1948. doi:10.2307/1969046.

20 Salvador Roura. Improved master theorems for divide-and-conquer recurrences. *Journal of the ACM (JACM)*, 48(2):170–205, 2001.