Multi-dimensional Boltzmann Sampling of context-free Languages

Olivier Bodini, Yann Ponty

Multi-dimensional Boltzmann Sampling of context-free Languages

Olivier Bodini[1] and Yann Ponty[2]

[1] Laboratoire d'Informatique de Paris 6 (LIP6), CNRS UMR 7606
Université Paris 6 - UPMC, 75252 Paris Cedex 05, France
[2] Laboratoire d'Informatique de l'École Polytechnique (LIX), CNRS UMR 7161/AMIB INRIA
École Polytechnique, 91128 Palaiseau, France

Abstract. This paper addresses the uniform random generation of words from a context-free language (over an alphabet of size k), while constraining every letter to a targeted frequency of occurrence. Our approach consists in an extended – multidimensional – version of the classic Boltzmann samplers [7]. We show that, under mostly *strong-connectivity* hypotheses, our samplers return a word of size in $[(1-\varepsilon)n,(1+\varepsilon)n]$ and exact frequency in $\mathcal{O}(n^{1+k/2})$ expected time. Moreover, if we accept a tolerance interval of length in $\Omega(\sqrt{n})$ for the number of occurrences of each letters, our samplers perform an approximate-size generation of words in expected $\mathcal{O}(n)$ time. We illustrate these techniques on the generation of Tetris tessellations with uniform statistics in the different types of tetraminoes.

1 Introduction

Random generation is the core of the simulation of complex data. It appears in real applicative domains such as complex networks (biology, Internet or social relationship), or software testing (validation, benchmarking). It helps us to predict the behavior of algorithms (complexities and statistical significance of results), to visualize limit properties (such as transition phases in statistical physics), to model real contexts (random graphs for web simulation).

Following the pioneering work of Flajolet *et al* [10], decomposable combinatorial classes can be specified using standard specifications. Two major techniques can then be applied to draw m objects of size n at random from such a class. On one hand, the recursive approach [14] precomputes the cardinalities of sub-classes for sizes up to n and uses these numbers to perform local choices that are consistent with the targeted uniformity. The best known optimization of this technique [5] uses certified floating point arithmetics and works in $\mathcal{O}(m{\cdot}n^{1+o(1)})$ but its implementation remains highly non-trivial due to its sophisticated precomputations. On the other hand, the Boltzmann sampling techniques, recently introduced by Duchon *et al* [7], achieves a random generation for most unlabelled [8] and labelled specifications in $\mathcal{O}(m\cdot n^2)$ operations at an optimally low $\mathcal{O}(m\cdot n)$ memory cost. Instead of enforcing a strict – and costly – control on the size of generated objects, this general technique rather induces an appropriate distribution on the size of sampled objects, and performs rejection until a suitable object is found.

In the present work, we investigate a natural multivariate extension of Boltzmann sampling, aiming at drawing objects, uniformly at random, having a prescribed composition in the different terminal letters. From a combinatorial perspective, such a generation allows the so-called symbolic method to reclaim combinatorial classes and languages that *fall slightly off* of its natural expressivity. For instance, restrictions of rational languages may not admit a rational (or even-context-free) specification under the additional hypothesis that some letters co-occur strictly (One may consider the triple-copy language). For context-free languages on k letters, this problem was previously addressed within the recursive framework [14] by Denise *et al* [5], deriving an algorithm in $\Theta(n^k)/\Theta(n^{2k})$ arithmetic operations respectively for rational and context-free languages. Using properties of holonomic series, Bertoni *et al* [3] revisited the problem and proposed a method for the uniform sampling from rational languages on two letters in $\Theta(n)$. Unfortunately a direct generalization of the technique only generalizes into a $\Theta(n^{k-1})$ algorithm for k letters, as pointed out in Radicioni's thesis [13].

Size Tolerance	Composition Tolerance	Average complexity
$\varnothing$	$\varnothing$	$\mathcal{O}(n^{2+k/2})$
$\varnothing$	$\Omega(\sqrt{n})$	$\mathcal{O}(n^2)$
$\mathcal{O}(n)$	$\varnothing$	$\mathcal{O}(n^{1+k/2})$
$\mathcal{O}(n)$	$\Omega(\sqrt{n})$	$\mathcal{O}(n)$

Table 1. Summary of our results for the generation of a word of length n over k letters in strongly connected context-free languages.

Following the general philosophy of Boltzmann sampling, our algorithm will first relax the compositional constraint by using non-uniform samplers to draw objects whose average composition is fine-tuned to match the targeted one, and perform rejection until an acceptable object is found. By acceptable, one understands that generated objects must feature prescribed size and composition, while tolerances may be allowed on both requirements. Our programme can then be summarized in the three following phases:

Phase I. Figure out a set of weights such that the expected composition matches the targeted one.
Phase II. Draw structures from a weighted distribution, using either the recursive approach (See [5]) or a weighted Boltzmann sampler (See section 4).
Phase III. Reject structures of unsuitable compositions, until an adequate object is generated and returned.

Although phases II and III are independently addressed in our analyses, one can (and will) combine them into a single rejection step when a weighted Boltzmann sampler is used for Phase II. The algorithmic aspects of our programme will essentially build on and extend previous works addressing the uniform version, but a general analysis of its overall performance is more challenging. Indeed, the complexity of the rejection Phase III is heavily related to the limiting (joint) distribution of the associated multivariate generating functions. For each phase, we attempt to give mathematical characterizations of classes having proper behaviors. In particular, for context free languages whose grammars are **strongly connected**, we obtain for each combination of tolerances, the complexities summarized in Table 1.

The plan of this paper follows the different phases : Section 2 defines the concepts and notations used throughout the paper. Section 3 explains how to tune efficiently the parameters such that the targeted composition matches the average behavior (Phase I). In Section 4, we discuss the complexity of Phase II, the number of rejections needed to reach a word of suitable size (or suitable approximate size). The complexity of the multidimensional rejection (Phase III) is addressed in Section 5. We illustrate our method in Section 6 by sampling perfect Tetris tessellations – tesselations of a $w \times h$ rectangles using balanced lists of tetraminoes. Finally we conclude with a short overview of future works.

2 Notations and definitions

Following traditional mathematical notations, we will use bold symbols for multi-dimensional variables/functions (i.e. $\boldsymbol{x}$), and use subscripts to access a specific dimension (i.e. x_i). Throughout the rest of the document, we will denote by Σ the **alphabet** of k letters, and by $\mathcal{C}$ a **context-free language** over Σ.

Composition and tolerance. Define the **composition** of sampled words as the **frequency** of occurrences of each letter t_i in a word $w \in \mathcal{C}$, denoted by $\boldsymbol{p}(w) := (|w|_{t_i}/n)_{i\in[1,k]}$. Our main goal is to generate – uniformly at random – some word $w \in \mathcal{C}$ having a composition that is *close to* a **targeted composition** $\boldsymbol{f} \in [0,1]^k$.

We make this notion of proximity explicit, and formalize the notion of acceptability for a sampled word. Namely let $\boldsymbol{\epsilon}$ be a k-tuple of positive real numbers and $\alpha \in \mathbb{Q}^+$ a rational exponent, an object

Epsilon	$\mathcal{C}=1$	$C_{\boldsymbol{\pi}}(z)=1$	$\Gamma C_{\pi}(x):=\varepsilon$
Letters	$\mathcal{C}=t_i$	$C_{\boldsymbol{\pi}}(z)=\pi_{t_i}z$	$\Gamma C_{\pi}(x):=t_i$
Union	$\mathcal{C}=\mathcal{A}+\mathcal{B}$	$C_{\boldsymbol{\pi}}(z)=A_{\boldsymbol{\pi}}(z)+B_{\boldsymbol{\pi}}(z)$	$\Gamma C_{\pi}(x):=\text{Bern}\left(\frac{A_{\boldsymbol{\pi}}(x)}{C_{\boldsymbol{\pi}}(x)},\frac{B_{\boldsymbol{\pi}}(x)}{C_{\boldsymbol{\pi}}(x)}\right)\longrightarrow \Gamma A_{\pi}(x)\mid \Gamma B_{\pi}(x)$
Product	$\mathcal{C}=\mathcal{A}\times\mathcal{B}$	$C_{\boldsymbol{\pi}}(z)=A_{\boldsymbol{\pi}}(z)\times B_{\boldsymbol{\pi}}(z)$	$\Gamma C_{\pi}(x):=\Gamma A_{\pi}(x).\Gamma B_{\pi}(x)$

Fig. 1. Weighted generating functions and weighted Boltzmann samplers for context-free languages.

$w\in\mathcal{C}$ qualifies as $(\boldsymbol{\epsilon},\alpha)$**-acceptable** if and only if

$$\boldsymbol{p}(w)_i\in I(f_i,\epsilon_i,\alpha),\ \forall i\in[1,k]$$

where $I(m,e,a):=[m-m^a e, m+m^a e]$. This definition captures the case of fixed (exact) compositions by setting $\alpha=1$ and $\epsilon_i=1/n,\forall i\in[1,k]$.

Weighted distributions. The following notions and definitions, recalled here for the sake of self-containment, can be found in Denise *et al* [5]. A positive **weight** vector $\boldsymbol{\pi}$ will assign positive weights $\pi_i\in\mathbb{R}^+$ to each letter $t_i\in\Sigma$. The weight is then extended multiplicatively on any object w by $\pi(w)=\prod_{x\in w}\pi_x$. This gives rise to the notion of **weighted generating function** $C_{\boldsymbol{\pi}}(z)$ for a context-free language $\mathcal{C}$, a natural generalization of the size (enumerative) generating function where each structure is counted with multiplicity equal to its weight

$$C_{\boldsymbol{\pi}}(z)=\sum_{w\in\mathcal{C}}\pi(w)z^{|w|}=\sum_{n\geq 0}c_{\pi,n}z^n$$

where $c_{\pi,n}$ is the total weight[3] of objects of size n. Notice that this generating function can be re-interpreted as a multivariate generating function in $\boldsymbol{\pi}$ and z

This weighting scheme implicitly defines a **weighted distribution** on the set $\mathcal{C}_n$ of words of size n, such that

$$\mathbb{P}(w\mid n)=\frac{\pi(w)}{\sum_{w'\in\mathcal{C}_n}\pi(w')}=\frac{\pi(w)}{c_{\pi,n}}.$$

Finally, the weighted distribution can be generalized into a **Boltzmann weighted distribution** on the whole language such that

$$\mathbb{P}_{x,\boldsymbol{\pi}}(w\mid n)=\frac{\pi(w)x^n}{\sum_{w'\in\mathcal{C}}\pi(w')x^{|w'|}}=\frac{\pi(w)x^n}{C_{\boldsymbol{\pi}}(x)}. \tag{1}$$

Property 1. Let N (resp. N_i) be the random variable for the size (resp. number of occurrences of a letter t_i) of any word in a $(x,\boldsymbol{\pi})-$ Boltzmann weighted distribution over a class $\mathcal{C}$. Then the expectations $\mathbb{E}_{x,\boldsymbol{\pi}}(N)$ and $\mathbb{E}_{x,\boldsymbol{\pi}}(N_i)$ are obtained from the partial derivatives of the multivariate generating function $C_{\boldsymbol{\pi}}(z)$ through

$$\mathbb{E}_{x,\boldsymbol{\pi}}(N)=x\frac{\frac{dC_{\boldsymbol{\pi}}(x)}{dx}}{C_{\boldsymbol{\pi}}(x)},\qquad \mathbb{E}_{x,\boldsymbol{\pi}}(N_i)=\frac{\pi_i\frac{\partial}{\partial\pi_i}C_{\boldsymbol{\pi}}(x)}{C_{\boldsymbol{\pi}}(x)} \tag{2}$$

In the sequel we will denote by $\boldsymbol{\mu}(x,\boldsymbol{\pi})$ the vector of expectations $(\mathbb{E}_{x,\boldsymbol{\pi}}(N_1),\cdots,\mathbb{E}_{x,\boldsymbol{\pi}}(N_k))$.

3 Tuning weights (Phase I)

First, let us address the question of finding a vector $\boldsymbol{\pi}$ such that the multidimensional rejection scheme (Phase III) is as efficient as possible. We propose and explore two alternatives, both computing a weights vector that make the expected and targeted compositions coincide. The first one uses a numerical Newton iteration. The second one uses an asymptotic approximation for the value of z which greatly simplifies the weights/frequencies relationship.

[3] This quantity is essentially similar to the partition function in statistical mechanics, dear to L. Boltzmann. . .

Input: Initial parameters z_0 and $\boldsymbol{\pi}_0$, a composition $\boldsymbol{f}$, a size n and ϵ a numerical precision
Output: The valid weights
Let $\mathbb{E}_{z_0}$ be the map from the space of the weights into $\mathbb{R}_+^k$ such that $\mathbb{E}_{z_0}(\boldsymbol{\pi}) = \boldsymbol{\mu}(z_0, \boldsymbol{\pi})$;
Let $J(\mathbb{E}_{z_0}(\boldsymbol{\pi}))$ be the Jacobian matrix of $\mathbb{E}_{z_0}(\boldsymbol{\pi})$;
$\boldsymbol{\pi} := \boldsymbol{\pi}_0$;
repeat
    end:=true; $\boldsymbol{c} := n\boldsymbol{f}$; $N := ||\boldsymbol{c} - \mathbb{E}_{z_0}(\boldsymbol{\pi})||$;
    while $N > \epsilon$ **do**
        $\boldsymbol{\pi}_{aux} := \boldsymbol{\pi}$;
        $\boldsymbol{\pi} := J(\mathbb{E}_{z_0})^{-1}(\boldsymbol{\pi}) \cdot (n\boldsymbol{f} - \mathbb{E}_{z_0}(\boldsymbol{\pi})) + \boldsymbol{\pi}$;
        if $N < ||\boldsymbol{c} - \mathbb{E}_{z_0}(\boldsymbol{\pi})||$ **then**
            $\boldsymbol{\pi} := \boldsymbol{\pi}_{aux}$; $\boldsymbol{c} := (\boldsymbol{c} + \mathbb{E}_{z_0}(\boldsymbol{\pi}))/2$; end:=false;
        end
    end
until *end=true*;
return $\boldsymbol{\pi}$

Algorithm 1: Tracking the weights.

Tuning by expectation. Newton's methods are based on successive linear (or higher order) approximations in order to obtain numerical estimates of a root of a system of equations. It is generally an efficient algorithm assuming that the initial values are close enough to a root. Here, we are interested in finding the unique root $(z_0, \boldsymbol{\pi_f})$ of the system $\boldsymbol{\mu}(z_0, \boldsymbol{\pi}) = n\boldsymbol{f}$. Algorithm 1 is a slightly revisited version of Newton's method which tests at each step if Newton's approximation has improved the estimate of the root. This test fails if and only if the current parameters are too far from the solution. In this case, we search using dichotomy an intermediate target that is closer to the solution than the current parameters.

Proposition 1. *Let $\boldsymbol{f}$ and n be the targeted composition and size respectively. Assume that the Jacobian matrix $J(\mathbb{E}_{z_0}(\boldsymbol{\pi_f}))$ is not singular*[4]*, then Algorithm 1 returns parameters $(z_0, \boldsymbol{\pi}_1)$ such that the expected composition $\boldsymbol{\mu}(z_0, \boldsymbol{\pi}_1)$ satisfies $||\boldsymbol{\mu}(z_0, \boldsymbol{\pi}_1) - n\boldsymbol{f}|| < \epsilon$.*
Moreover, there exists a neighborhood B of $(z_0, \boldsymbol{\pi_f})$ such that, for any $\boldsymbol{\pi}_0 \in B$, Algorithm 1 with initial condition $\boldsymbol{\pi}_0$ quadratically converges to $\boldsymbol{\pi_f}$ (i.e. $\exists C > 1$ such that $\forall k \geq 0, ||\boldsymbol{\pi}_k - \boldsymbol{\pi_f}|| \leq C^{-2k}$ where $\boldsymbol{\pi}_{k+1} := J(\mathbb{E}_{z_0})^{-1}(\boldsymbol{\pi}_k) \cdot (n\boldsymbol{f} - \mathbb{E}_{z_0}(\boldsymbol{\pi}_k)) + \boldsymbol{\pi}_k$).

Asymptotic tuning. Since one generally attempts to generate large objects, a natural option consists in solving the simpler asymptotic system.

Proposition 2. *Let us consider a context-free (resp. rational) language whose grammar is irreducible and aperiodic, and whose generating functions $C_{\boldsymbol{\pi}}(z)$ admits $\rho(\boldsymbol{\pi})$ as its dominant singularity of the generating function,*

$$\text{for any letter } t,\ \mathbb{E}_{z,\boldsymbol{\pi}}(N_t) \sim \frac{1}{2}\pi_t n \frac{\frac{\partial}{\partial \pi_t}\rho(\boldsymbol{\pi})}{\rho}\ \text{(resp. } \mathbb{E}_{z,\boldsymbol{\pi}}(N_t) \sim -\pi_t n \frac{\frac{\partial}{\partial \pi_t}\rho(\boldsymbol{\pi})}{\rho}\text{) as } z \text{ tends to } \rho(\boldsymbol{\pi}).$$

Remark 1. Considering the expectation $\mathbb{E}_n(N_t)$ of the number of letters t in a word of **fixed** size n. Then, from [5], similar asymptotic estimates holds for $\mathbb{E}_n(N_t)$ and the weights computed by our methods can therefore be used by the recursive approach.

4 Efficiency of the size rejection scheme (Phase II)

At this point, we assume that a k-tuple of weights $\boldsymbol{\pi}$ has been found such that the average composition in the weighted distribution matches the targeted one. We now need to perform a random generation of m words from the contex-free language with respect to the $\boldsymbol{\pi}$-weighted distribution.

[4] i.e. there is no linear dependency between the expected numbers of different letters.

Input: Parameters $x, \boldsymbol{\pi}$
Output: Object of $\mathcal{A}$ of size in $I(n,\varepsilon) := [n(1-\varepsilon), n(1+\varepsilon)]$
repeat
| $\gamma := \Gamma A_\pi(x)$
until $|\gamma| \in I(n,\varepsilon)$;
return (γ)

Algorithm 2: Rejection algorithm $\Gamma_2\mathcal{A}(x,\boldsymbol{\pi};n,\varepsilon)$

This problem was previously addressed in Denise *et al* [5] within the framework of the recursive method, and an algorithm in $\mathcal{O}(m \cdot n)$ arithmetic operations was proposed. Despite its apparent low complexity, the exponential growth of the numbers processed by the algorithm increases the practical complexity to $\Theta(m \cdot n^2)$ in time and $\Theta(n^2)$ in memory. Therefore we investigate a weighted generalization of the so-called Boltzmann sampling [7].

Weighted Boltzmann generation Let us remind that Boltzmann sampling first relaxes the size constraint and draws objects in a Boltzmann distribution of parameter x. To that purpose a fixed stochastic process, coupled with an (anticipated) rejection procedure, is used (See Figure 2). The probabilities of the different alternatives are precomputed by an external procedure called **oracle** (Symbolic algebra, or numerical method [12], see Appendices for details). A judicious choice of value for x ensures a low probability of rejection and this approach yields, for large classes of structures (Trees, sequences, runs, mappings, fountains...), generic algorithms in $\mathcal{O}(n^2)$ for objects of **exact-size** n, and in $\mathcal{O}(n)$ for objects of **approximate-sizes** in $[n(1-\varepsilon), n(1+\varepsilon)]$, for some $\varepsilon > 0$.

Through a minor modification of the oracle, one can easily turn unlabelled Boltzmann samplers introduced by Flajolet *et al* [8] into generators for the weighted Boltzmann distribution (See Eq. 1). Namely, one only needs to replace occurrences of the generating function $C(z)$ with its weighted counterpart $C_{\boldsymbol{\pi}}(z)$, obtaining the samplers summarized in Figure 2, and use the classic size rejection process (Algorithm 2).

Proposition 3. *Let $\boldsymbol{\pi}$ be a k-tuple of weights, x be a Boltzmann parameter, C be a context-free specification and $C_{\boldsymbol{\pi}}(z)$ its weighted generating function.*
Then any word $w \in \mathcal{C}$ is generated by the samplers summarized in Figure 2 with probability

$$\mathbb{P}_{x,\boldsymbol{\pi}}(w \mid n) = \frac{\pi(w)x^n}{C_{\boldsymbol{\pi}}(x)}.$$

The (renormalized) restriction of a $\boldsymbol{\pi}$-weighted Boltzmann distribution to objects of size n is clearly a $\boldsymbol{\pi}$-weighted distribution, and this fact ensures the correctness of a rejection-based approach.

Complexity analysis Let us qualify a context-free language as **well-conditioned** iff the singular exponent $\alpha_{\boldsymbol{\pi}}$ of its dominant singularity is non negative. It is then possible to restrict our analysis to such languages, associated with *flat* Boltzmann distribution, from the remark [7] that any language $\mathcal{C}$ can be pointed repeatedly, until its exponent becomes non-negative. Generating from the pointed version and *erasing* the point(s) generates any word of size n from $\mathcal{C}$ with respect to the weighted distribution.

Theorem 1 (Essentially proven in [7]). *Let $\mathcal{C}_{\boldsymbol{\pi}}$ be a weighted well-conditioned context-free language and x_n be the root in $(0, \rho_{\boldsymbol{\pi}})$ of $\mathbb{E}_{x,\boldsymbol{\pi}}(N) = n$, then*
a) The approximate-size sampler $\Gamma_2\mathcal{C}(x_n, \boldsymbol{\pi}; n, \varepsilon)$ has expected running time bounded by $\dfrac{\kappa n}{\zeta_{\alpha_{\boldsymbol{\pi}}}(\varepsilon)} + c(\boldsymbol{\pi})$.
b) The exact-size sampler $\Gamma_2\mathcal{C}(x_n, \boldsymbol{\pi}; n, 0)$ has expected running time bounded by $\dfrac{\kappa\Gamma(\alpha_{\boldsymbol{\pi}})n^2}{\alpha_{\boldsymbol{\pi}}^{\alpha_{\boldsymbol{\pi}}}} + c(\boldsymbol{\pi})n$.
where κ is the cost-per-letter induced by the canonical Boltzmann samplers, $\alpha_{\boldsymbol{\pi}}$ is the singular exponent of the dominant singularity of $C_{\boldsymbol{\pi}}(z)$, $\zeta_{\alpha_{\boldsymbol{\pi}}}(\varepsilon) = \dfrac{\alpha_{\boldsymbol{\pi}}^{\alpha_{\boldsymbol{\pi}}}}{\Gamma(\alpha_{\boldsymbol{\pi}})}\displaystyle\int_{-\varepsilon}^{\varepsilon}(1+s)^{\alpha_{\boldsymbol{\pi}}-1}e^{-\alpha_{\boldsymbol{\pi}}(1+s)}ds$, $\Gamma(x)$ is the gamma function, and $c(\boldsymbol{\pi})$ is homogeneous on n.

In particular, for any fixed weight vector $\boldsymbol{\pi}$, Theorem 1 implies a $\mathcal{O}(n)$ (resp. $\mathcal{O}(n^2)$) complexity for the approximate-size (resp. exact size) weighted samplers. Now enforcing through weights compositions that are unnatural ($\mathcal{O}(\sqrt{n})$ occurrence while naturally observed $\mathcal{O}(n)$ ones) may lead to a – somewhat hidden – dependency of $\boldsymbol{\pi}$ in n. Although we were unable to characterize these dependencies and their impact $c(\boldsymbol{\pi})$ on both complexities, we expect the latter to remain limited, and conjecture similar complexities when *meaningful* compositions (At least one occurrence of each letter) are targeted.

In the case of rational languages, the following theorem provides a computable evaluation for the efficiency of the size-rejection process. It relies on the partial fraction expansion of rational functions, which can be obtained for any weighted generating function $C_{\boldsymbol{\pi}}(z)$, and is denoted by

$$C_{\boldsymbol{\pi}}(z) = \sum_{i=1}^{r}\sum_{k=1}^{m_i}(1 - z/\rho_i)^{-\alpha_{i,k}} h_{i,k} + P(z) \tag{3}$$

where $P(z)$ is a polynomial of degree less than the number of states, r the number of distinct roots of $\det(\mathbb{I} - z \cdot \boldsymbol{M}) = 0$ and m_i the multiplicity of ρ_i which are sorted by increasing module. In weighted generating functions, ρ_i, $P(z)$, $h_{i,k}$, k and r depend on the actual values of the weights.

Theorem 2. *Let $\mathcal{C}_{\boldsymbol{\pi}}$ be a weighted rational language and x_n be the root in $(0, \rho_{\boldsymbol{\pi}})$ of $\mathbb{E}_{x,\boldsymbol{\pi}}(N) = n$, then the approximate-size sampler $\Gamma_2\mathcal{C}(x_n, \boldsymbol{\pi}; n, \varepsilon)$ succeeds in expected number of trials*

$$\frac{C_{\boldsymbol{\pi}}(x_n)}{\left(\sum_{i=1}^{r}\sum_{k=1}^{m_i}\binom{n+k-1}{k-1}(\rho_i)^{-n}h_{i,k} + [z^n]P(z)\right)(x_n)^n}.$$

5 Complexity of the multidimensional rejection (Phase III)

General principle Our approach relies on a rejection scheme that generalizes that of the classic – univariate – Boltzmann sampling. Words are drawn from a weighted distribution – rejecting those featuring a proportion of the parameters which too distant from the targeted one – until an acceptable one is found and returned. This gives the following rejection sampler $\Gamma_3\mathcal{A}(x, \boldsymbol{\pi}; n, \boldsymbol{m}, \varepsilon, \sigma)$ where x is real, $\boldsymbol{\pi}$ a real k-vector, $\boldsymbol{m}$ a map from $\mathbb{N}$ to $\mathbb{R}^k$, and ε the tolerance:

Input: The parameters $x, \boldsymbol{\pi}, n, \boldsymbol{m}, \varepsilon, \sigma$
Output: An object of $\mathcal{A}$ of size s in $I(n, \varepsilon)$
and for every parameter π_i, the number of occurrences of Z_i is in
$I(m_i(s), \varepsilon, \sigma) := [m_i(s) - m_i(s)^\sigma\varepsilon, m_i(s) + m_i(s)^\sigma\varepsilon]$
repeat
| $\gamma := \Gamma_2\mathcal{A}(x, \boldsymbol{\pi}; n, \varepsilon)$
until $\forall i, |\gamma|_i \in I(m_i(s), \varepsilon, \sigma)$;
return (γ)

Algorithm 3: $\Gamma_3\mathcal{A}(x, \boldsymbol{\pi}; n, \boldsymbol{m}, \varepsilon, \sigma)$

In many important classes of combinatorial structures, the composition of a random object is concentrated around its mean. It follows that a rejection-based generation can succeed after few attempts, provided that the expected composition matches the targeted one. Our main result is that, for any irreducible and simple context-free language, a suitably parameterized multidimensional rejection sampler generates a word of targeted composition after $\mathcal{O}(n^{k/2})$ attempts. Moreover, allowing a $\mathcal{O}(\sqrt{n})$ tolerance on the number of occurrences of each letters yields a sampler that succeeds in expected number of attempts asymptotically homogenous in n.

Now, let us denote by $U_n(\boldsymbol{\pi}_0)$ the k-multivariate random variable which follows the probability $\mathbb{P}(U_n(\boldsymbol{\pi}_0) = \boldsymbol{a}) = \frac{[z^n\boldsymbol{\pi}^{\boldsymbol{a}}]C_{\boldsymbol{\pi}}(z)\boldsymbol{\pi}_0^{\boldsymbol{a}}}{[z^n]C_{\boldsymbol{\pi}_0}(z)}$, i.e. the distribution of the parameters for objects of size n. Moreover, let us denote by $\boldsymbol{\mu}(n, \boldsymbol{\pi}_0)$ the mean-vector of $U_n(\boldsymbol{\pi}_0)$ and by $\boldsymbol{V}(n, \boldsymbol{\pi}_0)$ its variance-covariance matrix. If we do not have any strict correlation between the parameters, the matrix $\boldsymbol{V}(n, \boldsymbol{\pi}_0)$ is positive definite (and so, invertible). We can then define a norm as $||\boldsymbol{u}||_{\boldsymbol{V}^{-1}} = \sqrt{\boldsymbol{u}^T\boldsymbol{V}(n, \boldsymbol{\pi}_0)^{-1}\boldsymbol{u}}$. Now, let $\boldsymbol{V}$

be a positive definite matrix, we denote by $\kappa(\boldsymbol{V}) := \inf_{||\boldsymbol{u}||_\infty=1}\{||\boldsymbol{u}||_{\boldsymbol{V}}\}$, the infinum distance[5] from the unit sphere to the center of the Banach space.

Definition 1. *The* σ-concentrated condition *is defined as :*

$$\limsup_{n\to\infty} ||\boldsymbol{\mu}(n,\boldsymbol{\pi})||_\infty^\sigma \kappa(\boldsymbol{V}(n,\boldsymbol{\pi})^{-1}) = c > \sqrt{k}/\varepsilon$$

Theorem 3 (Approximate composition). *Let x_n and $\boldsymbol{\pi_a}$ be the solution of $\mathbb{E}_{x,\boldsymbol{\pi}}(N) = n$ and $\mathbb{E}_{x,\boldsymbol{\pi}}(N_i) = a_i$. The map $\boldsymbol{m}$ is defined as the $\boldsymbol{m}(s) = \mathbb{E}_{s,\boldsymbol{\pi_a}}(N_i)$ and assume that :*
i) the standardized version $(U_n(\boldsymbol{\pi_a}) - \boldsymbol{\mu}(n,\boldsymbol{\pi_a}))/\mathfrak{f}(n)$ admits a limiting distribution when n tends to the infinity,
ii) the σ-concentrated condition is verified for $\sigma \leq 1$.
Then the expected number of trials (of $\Gamma_2\mathcal{A}(x_n,\boldsymbol{\pi};n,\varepsilon)$) of the rejection sampler $\Gamma_3\mathcal{C}(x_n,\boldsymbol{\pi_a};n,\boldsymbol{m},\varepsilon,\sigma)$ is upper-bounded by

$$\sup_{s\in I(n,\varepsilon)} \frac{(\varepsilon\ \kappa(\boldsymbol{V}(s,\boldsymbol{\pi_a})^{-1})||\ \boldsymbol{\mu}(s,\boldsymbol{\pi_a})||_\infty^\sigma)^2}{(\varepsilon\ \kappa(\boldsymbol{V}(s,\boldsymbol{\pi_a})^{-1})||\ \boldsymbol{\mu}(s,\boldsymbol{\pi_a})||_\infty^\sigma)^2 - k}$$

which tends to a constant as $n \to \infty$.

Remark 2. The condition i) is just a condition of asymptotic non fluctuation of the distribution of the parameters according to the size.

Theorem 4 (Exact composition). *Assume that $(U_n(\boldsymbol{\pi_a}))$ admits a multidimensional Gaussian law with mean $\boldsymbol{\mu}$ and variance-covariance matrix $\boldsymbol{V}$ proportional to $f(n)$ as limiting distribution when n tends to the infinity, then the exact-composition rejection sampler $\Gamma_3\mathcal{C}(x_n,\boldsymbol{\pi_a};n,\boldsymbol{m},0,1)$ succeeds after an expected number of trials in $(2\pi)^{k/2}(\det(\boldsymbol{V}))^{1/2} = \mathcal{O}(f(n)^{k/2})$.*

Proof. Just notice that the probability to draw an exact composition corresponds to taking $\boldsymbol{u} = \boldsymbol{\mu}$ in the asymptotic estimate

$$p(\boldsymbol{u}) = \frac{1}{(2\pi)^{k/2}(\det(\boldsymbol{V})^{1/2}} \exp\Big(-\tfrac{1}{2}(\boldsymbol{u}-\boldsymbol{\mu})^t\ \boldsymbol{V}^{-1}(\boldsymbol{u}-\boldsymbol{\mu}) + o(1)\Big).$$

Consequently the expected number of attempts is $(2\pi)^{k/2}\det(\boldsymbol{V})^{1/2} = \mathcal{O}(f(n)^{k/2})$.

5.1 Rational languages: The Bender-Richmond-Williamson theorem [2]

The Bender-Richmond-Williamson theorem [2, Theorem 1] defines sufficient conditions such that the limiting distribution of a rational language $\mathcal{R}$ is a multidimensional Gaussian distribution. Let us remind that a rational language is **irreducible** if its minimal automaton $\mathcal{A}$ is strongly-connected, and **aperiodic** – if the cycle lengths in $\mathcal{A}$ have greatest common divisor equal to 1. Additionally the **periodicity parameter lattice** Λ, defined in [2] (Definition 2) is required to be full dimensional to avoid trivial correlations in the occurrences of letters.

Theorem 5. *Let $\mathcal{R}_{\boldsymbol{\pi}}$ be a weighted rational language whose minimal automaton is irreducible and aperiodic, and x_n be the root in $(0,\rho_{\boldsymbol{\pi}})$ of $\mathbb{E}_{x,\boldsymbol{\pi}}(N) = n$. Assume that the periodicity parameter lattice Λ is full dimensional then:*
a) For all $\sigma \geq 1/2$, the approximate-composition sampler $\Gamma_3\mathcal{R}(x_n,\boldsymbol{\pi};n,\varepsilon,\sigma)$ succeeds after $\mathcal{O}(1)$ trials
b) The exact-composition rejection sampler $\Gamma_3\mathcal{R}(x_n,\boldsymbol{\pi};n,0,1)$ succeeds after $\mathcal{O}(n^{k/2})$ trials.

Proof. From the system of language equations $\boldsymbol{\mathcal{L}} = \boldsymbol{M}\cdot\boldsymbol{\mathcal{L}} + \boldsymbol{\mathcal{E}}$, we directly obtain the system $\boldsymbol{L} = z\boldsymbol{M}\cdot\boldsymbol{L} + \boldsymbol{E}$ for the generating function. In this case the Peron-Frobenius theorem ensures that the dominating pole of every L_i in $\boldsymbol{L}$ is the least real value of $\det(\mathbb{I} - z\cdot\boldsymbol{M}) = 0$ and that this pole is simple. Now, assume that the *periodicity parameter lattice* Λ defined in [2] (Definition 2) is full dimensional.

[5] Recall that the infinity norm is defined as $||\boldsymbol{u}||_\infty = \max(|u_1|,\cdots,|u_k|)$

Assume also that we have a compact set Π_1 for the parameters in which the singular exponent is constant and equal to 1. Then from the Bender-Richmond-Williamson Theorem (see [2], Theorem 1 and [1]), it follows that for any fixed parameter in the compact set Π_1, the limiting distribution of the parameters is a multidimensional Gaussian distribution with mean and variance-covariance matrices proportional to n. Consequently, Theorem 3 applies for $\sigma = 1/2$, Theorem 4 applies with $f(n) = n$, and the result follows.

Let us discuss the relevance of the prerequisites of Theorem 5. If the matrix $\boldsymbol{M}$ is not aperiodic, there exists a power d such that $\boldsymbol{M}^d$ is aperiodic. So, we can always reduce the problem to a list of d aperiodic ones, and Theorem 5 applies under the same assumptions (full dimensional periodicity parameter lattice and compact set with constant singular exponents). The *irreducibility* requirement may be lifted when one of the strongly connected components dominates asymptotically, when the associated schema only involves subcritical and supercritical compositions (In the sense of [9, Theorem IX.2]). However the case of a competition between different components in a non irreducible automaton is much more challenging and requires serious developments that cannot be included in this short paper. Finally we point out that, with minor modifications, similar results could be obtained for more general transfer matrix models.

5.2 Context-free languages: Drmota's theorem [6]

A theorem by Drmota [6] gives very similar sufficient conditions for the limiting multivariate distribution to satisfy the conditions of Theorem 4. Namely, the irreducibility condition needs be fulfilled by the **dependency graph** of the grammar – the directed graph on non-terminals whose edges connect left hand sides of rules to their associated right-hand sides. The lattice and aperiodicity properties are replaced by the very similar concept of **simple type** grammar, requiring the existence of a *positive* $k+1$ dimensional cone centered on $\mathbf{0}$ in the space of coefficients.

Theorem 6. *Let $\mathcal{C}_{\boldsymbol{\pi}}$ be a weighted context-free language having a grammar $\mathcal{G}$, and x_n be the root in $(0, \rho_{\boldsymbol{\pi}})$ of $\mathbb{E}_{x,\boldsymbol{\pi}}(N) = n$. Assuming that $\mathcal{G}$ is of simple-type ([6, Theorem 1]) and its dependency graph is strongly connected then for all $\sigma \geq 1/2$, then the complexities of Theorem 5 also hold for $\mathcal{C}_{\boldsymbol{\pi}}$.*

Again, the strong-connectedness requirement could be relaxed for disconnected grammars whose behavior is dominated by that of a single connected component. A formal characterization of such grammars may borrow to the theory of (sub/super)-critical compositions (See [9, Theorem IX.2]).

6 Sampling perfect Tetris histories

In this short illustration, we address the generation of Tetris tessellations, i.e. tessellations using tetraminoes of a board having prescribed width w. The Tetris game consists in placing falling tetraminoes (or **pieces**) $\mathcal{P}$ in a $w \times h$ board. The goal of the player is to create hole-free horizontal lines which are then eliminated, and the game goes on until the pieces stack past the ceiling of the board. Most implementations of Tetris use the so-called *bag strategy*, which consists in giving the player sequences of permutations of the 7 types of tetraminoes, therefore inducing an even composition in each tetramino type. A rational specification (Built by Algorithm 4) exists for Tetris tessellations of any fixed width, but the additional constraint on composition provably throws the associated language out of the context-free class. Therefore, we choose to model the generation of evenly-distributed Tetris tessellations as a multivariate generation from a rational language. Such tessellations could in turn be used as a basic construct to build hard instances for the offline version of the algorithmic Tetris problems [4,11].

6.1 Building the automaton of Tetris tesselations

First let us find an unambiguous decomposition of Tetris tessellations. The idea is to focus on the state of the upper band of the tessellation that is not entirely filled, or **boundary** of a tessellation.

Input: The board width w and the flat boundary $\mathcal{B}_w$
Output: Q the states set and σ the transition function of $\mathcal{A}_w = (\mathcal{P}, Q, \mathcal{B}_w, \{\mathcal{B}_w\}, \sigma)$

```
begin
    (Q, σ) ← (B_w, ∅)
    S ← {B_w}
    while S ≠ ∅ do
        S ⇒_pop B;
        for p ∈ P_B do
            B′ ← B − p;
            if B′ ∉ Q then
                Q ← Q ∪ {B′};
                S ⇐_push B′;
            end
            σ ← σ ∪ {(B, p, B′)};
        end
    end
    return (Q, σ)
end
```

Width w	#States in $\mathcal{A}_w$	#States minimal
2	4	minimal
3	55	minimal
4	80	78
5	1686	1646
6	4247	4130
7	41389	40099
8	49206	47564
9	919832	–

Algorithm 4: Constructing the automaton $\mathcal{A}_w$ for tessellations of width w. Right: Growth of the number of states for increasing values of w.

In particular for tessellations the upper band is completely filled and the associated boundary is **flat**. One can investigate the different ways to get to a given boundary $\mathcal{B}$ by simulating the adjunction of a piece p to another boundary $\mathcal{B}'$, or conversely its **removal** from $\mathcal{B}$, which we favor in the following. Without further restriction on the position of removal, such a decomposition would be *ambiguous* and give rise to an infinite number of different boundaries. Consequently, we enforce a canonical order on the removal of pieces by restricting it to a set of (possibly rotated) pieces $\mathcal{P}_\mathcal{B}$ positioned such that the upper-rightmost position of the piece matches that of the boundary, and the piece is entirely contained in the boundary. We refer to the induced decomposition as the **disassembly decomposition**.

Proposition 4. *The disassembly decomposition generates sequences of removals from and to flat boundaries that are in bijection with Tetris tesselations.*

Proof (Sketch of). Let us discuss briefly the correctness of this decomposition, or equivalent that the sequences of k removals leading from a **flat boundary** $\mathcal{B}_w$ to itself are in bijection with the tessellations of $w \times 4 \cdot k/w$. First let us notice that the decomposition is unambiguous, since all the local removals share at least one position (the upper-rightmost of the boundary) and are therefore strongly ordered. Furthermore, it is also provably complete by induction on the number of piece n, since any tessellation has a upper-rightmost position which, upon removal, gives another tessellation of smaller size, and completeness of the decomposition propagates from tessellations of size n to size $n+1$. Finally, it gives rise to a finite number of states since the difference between the highest and lowest point in any reached boundary does not exceed the maximal height of a piece.

The finiteness of the state space yields Algorithm 4 that builds the automaton $\mathcal{A}_w$, generating tessellations of width w. Notice that the resulting automaton in not necessarily co-accessible, since the removal of some piece can create boundaries that cannot be completed into a flat one through any sequence of removal. Consequently, we added in our implementation a test of connectedness that discards any boundary having a (dis)connected component involving a number of blocks that is not a multiple of 4, as such boundaries clearly cannot reach a flat state again. Running a minimization algorithm of the resulting automata confirms the expected explosion in the number of states (See Algorithm 4) required for increasing values of w.

6.2 Random generation

First we point out that the automaton has matching initial and final states, so the strong connectedness is obviously ensured and our theorems regarding the complexity of generation apply. One can then translate it into a system of functional equations involving the (rational) generating functions associated with each states. Solving the system gives the generating functions, from which one can extract many informations.

For instance, fixing the width $w = 6$ and a number $n = 105$ of pieces, one obtains a number $h_{6,105} = 3.10^{71}$ of potential tessellations, and extracting coefficients of derivatives (See Equation 2) yields:

Piece							
Frequency (%)	7.90	10.55	20.42	20.42	17.00	7.90	15.81

Consequently, the average composition of a Tetris tessellation is incompatible with the *bag strategy*, which results in instances that lead to evenly distributed pieces. One can then use the method described in Section 3 to compute a set of weights ensuring, on the average, one piece out of seven of each type, and get the following weight vectors

Piece							
Weight	0.939919	0.80	0.37561	0.373549	0.45759	0.957851	0.420387

A weight random generation for the $w = 6$ and $n = 105$, coupled with a rejection that allows the numbers of any piece to be equal to 15 ± 1, gives the instances drawn in Figure 3.

6.3 From random Tetris tessellations to Tetris instances

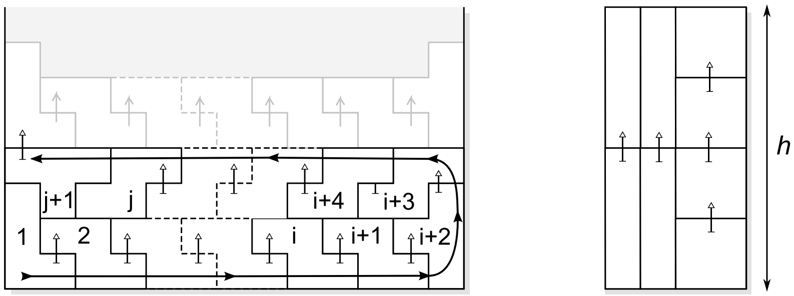

Fig. 2. Left: Tetris tessellations associated with a unique instance. Only the most relevant dependency points are displayed here (arrows) and pieces are labelled with their rank in the only compatible instance. Duplicating the gadget preserves the uniqueness of the associated instance while allowing for the generation of tessellations of arbitrarily large dimensions. **Right:** Tesselation realized by $\binom{h}{h/2} \in \Theta(2^n/\sqrt{n})$ different instances.

Proposition 5. *For any Tetris tesselation $\mathcal{T}$, there exists an instance (sequence of pieces) such that $\mathcal{T}$ can be obtained.*

Proof. Let us assume that $\mathcal{T}$ is a tessellation of a $w \times n$ rectangle using tetraminoes, and let us call *dependency point* any contact between the southward face of a piece $\mathcal{B}_1$ and the northward face of a piece $\mathcal{B}_2$. Such a point induces a directionality $\mathcal{B}_1 \to \mathcal{B}_2$, corresponding to the necessity of placing $\mathcal{B}_1$ before $\mathcal{B}_2$. This defines a *dependency graph* $D = (V, E)$ whose vertices are the pieces $\mathcal{B} \in \mathcal{T}$, and whose directed edges are such that $(\mathcal{B}_1, \mathcal{B}_2) \in E$ iff $\mathcal{B}_1 \to \mathcal{B}_2$. Additionally, each edge is labelled with the coordinate of its associated dependency point.

It can be shown that D is acyclic, by first proving that any path along D is labelled with coordinates that are either increasing on the y-axis or monotonic on the x-axis. Let us start by noticing that, aside from the and pieces, all types of pieces exhibit northward faces that are strictly higher than their southward ones. Furthermore, any heterogenous pair of piece only exposes northward faces that are at greater y-coordinates than their dependency point, inducing an increase of y-coordinate in the path. Consequently, there only exists two configurations of dependent pieces $A \to B$, namely

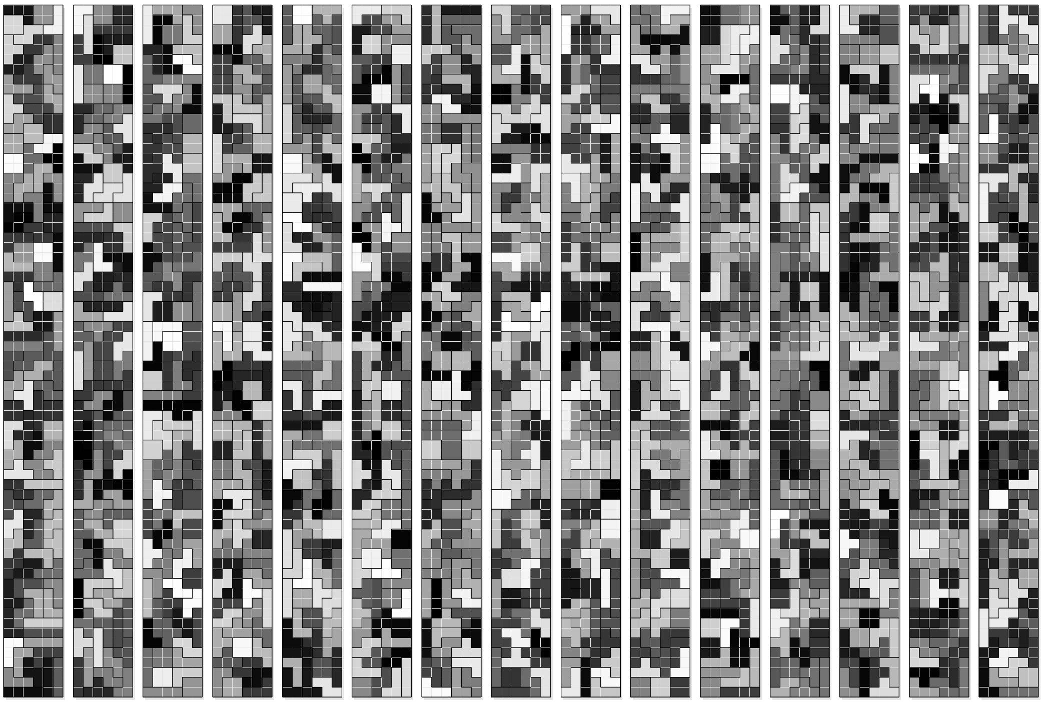

Fig. 3. Fifteen Tetris tesselations of width 6 having even composition (+/- 1) in the different pieces.

and , such that B exposes a southward face at the same height as their dependency point. The only way for a path in D not to increase in y-coordinate is then to feature a sequence of (resp.) pieces, inducing a monotonic behavior which proves our claim, and the acyclic nature of D follows.

Finally, the acyclicity of D implies the existence of a sequence of pieces realizing $\mathcal{T}$, since it is possible, in any configuration resulting in a possibility to remove a *free piece* at maximal y-ordinate. The reversal of any sequence of such pieces yields an instance such that $\mathcal{T}$ can be obtained.

There are a few limitations induced by our tesselation model. First it can be remarked that our notion of perfect histories does not capture every possible Tetris game ending with an empty board, as one may temporarily leave *holes* which amount to disconnecting pieces in the tesselation representation. Secondly, although there exists in the general case, many different free pieces to choose from while rebuilding an instance, there exists tesselations giving rise to a unique instance, as illustrated in Figure 2. Consequently, using the DAGs associated with Tetris histories to draw instances of the offline version of Tetris algorithmic problems [4] would favor exponentially certain instances over others. However, we still believe our method could be used to generate hard instances with the guarantee of a possible success which could be used to benchmark the performances of heuristics and exponential-time solutions.

7 Conclusion

In this paper, we have adapted and applied a general methodology for the multivariate random generation of combinatorial objects. Under explicit and natural conditions, random generators having complexity in $\mathcal{O}(n^{2+k/2})$ have been derived for the exact size and composition generation, outperforming best known algorithms (in $\mathcal{O}(n^k)$ and $\mathcal{O}(n^{2k})$ respectively for rational and context-free languages) for this problem. Furthermore, provided a small (linear) tolerance is allowed on the size of generated objects, and a $\Omega(n^{1/2})$ one is allowed in the other dimensions, our generators generate objects in linear expected time. We have applied these principles to the generation of perfect Tetris tessellations with uniform statistic in tetraminoes, which we have generated and drawn.

This paper is the first step toward a general analysis of the multi-parameter Boltzmann sampling. Compared to its alternative using the recursive method, the resulting method is not only theoretically faster, but also only requires only $\mathcal{O}(n)$ storage and its time complexity seems less affected by larger specifications. Nevertheless, many questions are left open, for instance with respect to the nature of the dependency between the weights and *reasonable* frequencies, which would allow us to address the complexities of Phase 2 in a much more general setting. The present work also implicitly assumes, at the multivariate rejection phase, that suitable weights have been found which is not always possible if the targeted distribution is incompatible with some dependencies in the grammar. A future direction might consist in investigating non-trivial, sufficient – yet tight – conditions such that the targeted composition can be achieved on the average.

Since multivariate Boltzmann samplers can be obtained in any situation where the distribution is well-concentrated, one may envisioned many classes, including constrained trees, permutations with a fixed number of cycles, functional graphs with a controlled number of components. . . A first step in this direction would consists in extending to simple Polya operators some of the multivariate theorems. As mentioned earlier in the document, the requirement of strong-connectedness (or irreducibility) could be questioned in the light of (sub/super)-critical compositions. Another similar direction is the use of Hwang's Quasi-powers theorem, giving rise to low variance distributions, for a general treatment of the bivariate case.

Acknowledgements

Olivier Bodini and Yann Ponty were funded by the ANR-GAMMA grant of the French *Agence-Nationale-Recherche.*

References

1. E. Bender and L. B. Richmond, *Central and local limit theorems applied to asymptotic enumeration II: Multivariate generating functions*, JCT serie A **34** (1983), 255–265.
2. E. Bender, L. B. Richmond, and S.G. Williamson, *Central and local limit theorems applied to asymptotic enumeration III: Matrix recursions*, JCT serie A **35** (1983), 263–278.
3. A. Bertoni, P. Massazza, and R. Radicioni, *Random generations of words in regular languages with fixed occurrences of symbols.*, Proceedings of Words'03, vol. 27, TUCS Gen. Publ., 2003, pp. 332–343.
4. Ron Breukelaar, Erik D. Demaine, Susan Hohenberger, Hendrik Jan Hoogeboom, Walter A. Kosters, and David Liben-Nowell, *Tetris is hard, even to approximate*, International Journal of Computational Geometry and Applications **14** (2004), no. 1–2, 41–68.
5. A. Denise, O. Roques, and M. Termier, *Random generation of words of context-free language saccording to the frequencies of letters*, Mathematics and Computer Science: Algorithms,Trees, Combinatorics and probabilities (D. Gardy and A. Mokkadem, eds.), Trends in Mathematics, Birkhaüser, 2000, pp. 113–125.
6. M. Drmota, *Systems of functional equations*, Random Structures and Algorithms **10** (1997), no. 1-2, 103–124.
7. P. Duchon, P. Flajolet, G. Louchard, and G. Schaeffer, *Boltzmann samplers for the random generation of combinatorial structures*, Combinatorics, Probablity, and Computing **13** (2004), no. 4–5, 577–625, Special issue on Analysis of Algorithms.
8. P. Flajolet, E. Fusy, and C. Pivoteau, *Boltzmann sampling of unlabelled structures*, Proceedings of ANALCO'07 (Analytic Combinatorics and Algorithms) Conference (SIAM Press, ed.), 2007.
9. Philippe Flajolet and Robert Sedgewick, *Analytic combinatorics*, Cambridge University Press, 2009.
10. Philippe Flajolet, Paul Zimmerman, and Bernard Van Cutsem, *A calculus for the random generation of labelled combinatorial structures*, Theoretical Computer Science **132** (1994), no. 1-2, 1–35.
11. Hendrik Jan Hoogeboom and Walter A. Kosters, *Tetris and decidability*, Inf. Process. Lett. **89** (2004), no. 6, 267–272.
12. C. Pivoteau, B. Salvy, and M. Soria, *Boltzmann oracle for combinatorial systems*, Algorithms, Trees, Combinatorics and Probabilities, Discrete Mathematics and Theoretical Computer Science, 2008, Proceedings of the Fifth Colloquium on Mathematics and Computer Science. Blaubeuren, Germany. September 22-26, 2008, pp. 475–488.
13. Roberto Radicioni, *Holonomic power series and their applications to languages*, Ph.D. thesis, Facoltà di scienze matematiche, fisiche et naturali, Università degli studi di Milano, 2006.

14. H. S. Wilf, *A unified setting for sequencing, ranking, and selection algorithms for combinatorial objects*, Advances in Mathematics **24** (1977), 281–291.

8 Annexes

8.1 Oracle computation

In the Boltzmann method, a crucial point consists in evaluating the generating function in the fixed tuned parameters. This could be computationally expensive, in particular when the generating function is not defined by a closed-formed expression but only by a system S of functional equations : $\boldsymbol{F}(Z,\pi) = \boldsymbol{\Phi}(\boldsymbol{F}, Z, \pi)$. First, we can say that this system is *well-founded* if the sequence of formal series $\boldsymbol{F}_0 = \mathbf{0}$ and $\boldsymbol{F}_n(Z,\pi) = \boldsymbol{\Phi}(\boldsymbol{F}_{n-1}, Z, \pi)$ is convergent[6] and its limit $\boldsymbol{F}_\infty$ is solution of S. The naïve way to evaluate series is just to following the sequence, not with formal parameters, but with real parameters. It is straightforward that if the parameters are taken in the convergence domain of the series, then the sequence converges to the evaluation of series in it (by observing that the tail of the serie becomes negligible).

Nevertheless this convergence is not very fast since only one digit is typically gained per iteration. As explained in the paper of Salvy *et al* [12], it is possible for univariate generating function to improve the speed of convergence by using classical Newton's method. They only restrict the domain of application to functional systems build by finite compositions of simple or analytic operators (as $+$, $\times$, $\frac{1}{1-x}$, e^x, $\ln(1/(1-x))$). An important point in [12] consists in proving that a system S is well-founded if and only if the Jacobian matrix $\partial\boldsymbol{\Phi}/\partial\boldsymbol{F}(\mathbf{0},0)$ is nilpotent (and consequently $\mathrm{Id} - \partial\boldsymbol{\Phi}/\partial\boldsymbol{F}$ is invertible).

Now, without any difficulty, their approach can be extended to multivariate generating functional systems. Indeed, let $\boldsymbol{F} = \boldsymbol{\Phi}(\boldsymbol{F}, Z, \boldsymbol{\pi})$ be a well-founded functional system with $\boldsymbol{\Phi}(\mathbf{0}, 0, \pi) = \mathbf{0}$. Let (x_0, π_0) be inside the domain of convergence of the generating series $\boldsymbol{F}(Z,\pi)$. Then the following iteration converges to $\boldsymbol{F}(x_0, \pi_0)$:

$$\boldsymbol{F}_0 = \mathbf{0},\ \boldsymbol{F}_{n+1} = \boldsymbol{F}_n + (\mathrm{Id} - (\partial\boldsymbol{\Phi}/\partial\boldsymbol{F})(\boldsymbol{F}_n, x_0, \pi_0))^{-1} \times (\boldsymbol{\Phi}(\boldsymbol{F}_n, x_0, \pi_0) - \boldsymbol{F}_n).$$

Now, in practice, we continue the iteration process until $||\boldsymbol{F}_{n+1} - \boldsymbol{F}_n|| < \varepsilon$ for a fixed arithmetical precision ε.

8.2 Proofs

Proof (Proposition 1.).

Let $\boldsymbol{\pi}$ be the current weight vector, the vector $\boldsymbol{v} = n\boldsymbol{f} - \mathbb{E}_{z_0}(\boldsymbol{\pi})$ (where $\boldsymbol{f}$ is the composition vector) indicates the direction of the decay. So, we could take as new current vector $\pi_c = J(\mathbb{E}_{z_0})^{-1}(\boldsymbol{\pi}) \cdot (n\boldsymbol{f} - \mathbb{E}_{z_0}(\boldsymbol{\pi})) + \boldsymbol{\pi}$. If, at each step, we stay inside the "combinatorial domain of the weights (i.e. the domain where the generating function is analytic), then the sequence converges to the solution of the system. To stay in the combinatorial domain : for each new π_c, we compare the norm $n_c = ||n\boldsymbol{f} - \mathbb{E}_{z_0}(\pi_c)||$ with $n = ||n\boldsymbol{f} - \mathbb{E}_{z_0}(\boldsymbol{\pi})||$. If $n_c > n$ the target can not be approximate directly. So, we add a new intermediate target $c = (n\boldsymbol{f} + \mathbb{E}_{z_0}(\boldsymbol{\pi}))/2$ and try to solve recursively this intermediate problem. If we have solved this new problem, the new current weights are closer from the targeted ones and we can try to solve the initial problem but with these new current weights.

The quadratic convergence is a classical property of the newton's method.

Proof (Proposition 2.).

By Perron-Frobenius theorem, assuming that $\rho(\boldsymbol{\pi})$ is the dominant singularity of the system $\boldsymbol{L}$, the generating function vector $\boldsymbol{L}$ verifies $\boldsymbol{L} \sim (1 - \frac{z}{\rho})^{-1}\boldsymbol{H}$ when $z \sim \rho$ where $\boldsymbol{H}$ is a functional matrix which are not singular in $|z| \leq \rho$. So, $\frac{\pi_l \frac{\partial}{\partial \pi_l} L_i}{L_i} \sim \pi_l (1 - \frac{z}{\rho})^{-1} z \frac{\frac{\partial}{\partial \pi_l}\rho(\boldsymbol{\pi})}{\rho^2}$. In particular, by taking $z = \rho(1 - \frac{1}{n})$, we obtain $\mathbb{E}_{z,\boldsymbol{\pi}}(N_t) \sim -\pi_t n \frac{\frac{\partial}{\partial \pi_t}\rho(\boldsymbol{\pi})}{\rho}$. Now, the system $f_t = \frac{n_t}{n} = -\pi_t \frac{\frac{\partial \rho(\boldsymbol{\pi})}{\partial \pi_t}}{\rho}$ for t in the alphabet can be solved with less difficulty than the initial one.

[6] The distance in the metric ring $\mathbb{R}[[Z]]$ is defined as $d(F,G) = 2^{-k}$ where k is the least integer such that the k-th coefficient of the two series F and G are different

Proof (Proposition 3.). Immediate from the proof of [8], Theorem 1.1:
$\mathcal{C} = 1$ *(Empty structure)* or $\mathcal{C} = t_i$ *(Atom)*: For singleton classes, equation 1 simplifies to 1 and both samplers draw the structure unconditionally.
$\mathcal{C} = \mathcal{A} + \mathcal{B}$ *(Union)*: Assuming the validity of both $\Gamma A_\pi(x)$ and $\Gamma B_\pi(x)$, the probability of sampling a structure w from $\mathcal{A}$ (resp. $\mathcal{B}$) is $\frac{A_{\boldsymbol{\pi}}(x)}{C_{\boldsymbol{\pi}}(x)}\frac{\pi(w)x^{|w|}}{A_{\boldsymbol{\pi}}(x)} = \frac{\pi(w)x^{|w|}}{C_{\boldsymbol{\pi}}(x)}$ (resp. $\frac{\pi(w)x^{|w|}}{C_{\boldsymbol{\pi}}(x)}$).
$\mathcal{C} = \mathcal{A} \times \mathcal{B}$ *(Product)*: Assuming the validity of both $\Gamma A_\pi(x)$ and $\Gamma B_\pi(x)$, each structure $w = w_\mathcal{A} w_\mathcal{B} \in \mathcal{C}$ is sampled with probability

$$\frac{\pi(w_\mathcal{A})x^{|w_\mathcal{A}|}}{A_{\boldsymbol{\pi}}(x)}\frac{\pi(w_\mathcal{B})x^{|w_\mathcal{B}|}}{B_{\boldsymbol{\pi}}(x)} = \frac{\pi(w_\mathcal{A})\pi(w_\mathcal{B})x^{|w_\mathcal{A}|+|w_\mathcal{B}|}}{A_{\boldsymbol{\pi}}(x)B_{\boldsymbol{\pi}}(x)} = \frac{\pi(w)x^{|w|}}{C_{\boldsymbol{\pi}}(x)}.$$

Proof (Theorem 3.). The proof is based on the multivariate Chebyshev inequality : Let X be an N-dimensional random variable with mean μ and covariance matrix V, then

$$\Pr\left(||X - \mu||_{V^{-1}} < t\right) > 1 - \frac{N}{t^2}.$$

Indeed, the probability to lie in the interval $I(m_i(s), \varepsilon, \alpha)$ can be rewritten as

$$\Pr\left(||U_s(\pi_{\boldsymbol{a}}) - \mu(s, \pi_{\boldsymbol{a}})||_\infty < \varepsilon||\mu(s, \pi_{\boldsymbol{a}})||_\infty^\alpha\right).$$

Now, we have the equivalent norm formula $||x||_\infty \leq \frac{1}{\kappa(V(s,\pi_{\boldsymbol{a}})^{-1})}||x||_{V(s,\pi_{\boldsymbol{a}})^{-1}}$. So, put $\mu = \mu(s, \pi_{\boldsymbol{a}})$ and $V^{-1} = V(s, \pi_{\boldsymbol{a}})^{-1}$, one gets

$$\Pr\left(||U_s(\pi_{\boldsymbol{a}}) - \mu||_{V^{-1}} < \varepsilon\ \kappa(V^{-1})||\mu||_\infty^\alpha\right) \leq \Pr\left(||U_s(\pi_{\boldsymbol{a}}) - \mu||_\infty < \varepsilon\ ||\mu||_\infty^\alpha\right).$$

By taking $t = \varepsilon\ \kappa(V^{-1})||\mu(s, \pi_{\boldsymbol{a}})||_\infty^\alpha$ in the Chebyshev inequality, we obtain

$$\Pr\left(||U_s(\pi_{\boldsymbol{a}}) - \mu(s, \pi_{\boldsymbol{a}})||_\infty < \varepsilon||\mu(s, \pi_{\boldsymbol{a}})||_\infty^\alpha\right) > 1 - \frac{k}{(\varepsilon\kappa(V^{-1})||\mu(s, \pi_{\boldsymbol{a}})||_\infty^\alpha)^2}.$$

The theorem ensues from the fact that the expected time to reach a good answer is $1/p$ when p is the probability to obtain it.