Multi-Parameter Sampling for Rational Languages

Olivier Bodini, Yann Ponty

Multi-Parameter Sampling for Rational 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 Algorithmique: Fondements et Applications (LIAFA), CNRS UMR 7089,
Université Paris Diderot, 75205 Paris Cedex 13, France

Abstract. This paper addresses how to efficiently sample words from a rational language (over an alphabet of size k), while constraining every letter to a targeted frequencies of occurrence. Our approach consists in an extended – multivariate – version of the classical Boltzmann samplers [6]. We prove that, under relatively weak hypotheses, our sampler returns 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 $\mathcal{O}(\sqrt{n})$ for the number of occurrences of each letters, our sampler reaches an approximate-size generation of words in expected $\mathcal{O}(n)$ time. We illustrate these techniques on the generation of perfect Tetris histories (Tesselations of a $w \times h$ rectangle).

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* [8], 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 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 delicate because of sophisticated precomputations. On the other hand, the Boltzmann sampling techniques, recently introduced by Duchon *et al* [6], achieves a random generation for most unlabelled [7] 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. For rational languages on k letters, this problem was previously addressed through the so-called recursive approach [12] by Denise *et al* [5] deriving an algorithm in $\Theta(n^k)$ arithmetic operations. 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 the technique involved in the optimization generalizes poorly to a $\Theta(n^{k-1})$ algorithm, as was pointed out in Radicioni's thesis [11].

From a combinatorial perspective, such a generation allows the so-called symbolic method to reclaim combinatorial classes and languages that *fall off* of its natural expressivity. We illustrate this general claim with a generation of perfect Tetris games, tessellations of a $w \times h$ boards using tetraminoes while conserving a uniform statistics on pieces. This latter constraint arises from the so-called *bag strategy* used for the random generation of sequences of pieces by most implementations of the Tetris game. This strategy consists in drawing sequences of permutations of the 7 types of tetraminoes, therefore inducing an equal compositions in each type of piece every 7 pieces. Consequently, the *language of perfect Tetris games* can be seen as an unordered (constrained) version of

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

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

a 7-copy language and is clearly not rational. However a rational specification can be found for any family of perfect games having a prescribed constant width w. Such specification can then be used in conjunction with a control of the number of occurrences in each piece to draw perfect games uniformly at random. Such games can in turn be used to build sequences of pieces, i. e. instances of the offline version of the algorithmic Tetris problems [4,9].

Following the philosophy underlying Boltzmann sampling, we first relax the compositional constraint by using non-uniform samplers and draw objects whose average composition is fine-tuned to match the targeted one. Then we 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 for 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 a 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 if 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 give mathematical characterizations of classes having proper behaviors. In particular, for rational languages whose automata 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 histories – tesselations of a $w \times h$ rectangles using 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 $\mathcal{C}$ be a **rational 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** $\boldsymbol{f} \in [0,1]^k$.

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 rational languages.

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 $w \in \mathcal{C}$ 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 rational 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

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

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

Rational generating functions of rational languages Let us consider the weighted generating function $L_{\boldsymbol{\pi}}$ of a rational language $\mathcal{L}$. It is a classic result of language theory that any rational language $\mathcal{L}$ is recognized by a deterministic minimal automaton. Such an automaton can in turn be transformed into a left-linear grammar generating exactly $\mathcal{L}$.

From the deterministic nature of the automaton, the resulting grammar is unambiguous and only uses the Cartesian product and disjoint union operators, respectively denoted by $\times$ and $+$ in the so-called symbolic method. One can therefore systematically translate such a grammar, using rules summarized in Table 1, into a system of functional equations involving the weighted generating function associated with each state. The system can be expressed using the following classical matrix representation $\boldsymbol{\mathcal{L}} = \boldsymbol{M} \cdot \boldsymbol{\mathcal{L}} + \boldsymbol{\mathcal{E}}$, where $\boldsymbol{\mathcal{E}}$ is the final state $\{0, \epsilon\}$-vector and the matrix $\boldsymbol{M}$ is the **transition matrix** of $\mathcal{A}$.

The resulting system is linear in each of the generating function associated with non-terminals, and can therefore always be solved by a simple Gaussian elimination. By considering the non-terminal/generating function associated with the initial state, one obtains the weighted generating function $L_{\boldsymbol{\pi}}$ of $\mathcal{L}$. Since this function is the solution of a linear system, it can be written as a fraction of polynomials in z and $\boldsymbol{\pi}$, i. e. a **rational generating function**.

3 Tuning weights (Phase I)

The weighted distribution impacts the average composition of a randomly generated word. Consequently, we address here the question of finding a vector $\boldsymbol{\pi}$ such that the multidimensional rejection scheme (Phase II) is as efficient as possible. We propose and explore two alternatives, both computing a weights vector making 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.

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 $(z_0, \boldsymbol{\pi})$ 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 $\boldsymbol{f}$ be a targeted composition and n be a size, the Algorithm 1 returns parameters z_0 and $\boldsymbol{\pi}$ such that the expected composition $\boldsymbol{\mu}(z_0, \boldsymbol{\pi})/n$ is ϵ-close to $\boldsymbol{f}$.*

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

Proposition 2. *Let us consider that the automaton is irreducible and aperiodic and that $\rho(\boldsymbol{\pi})$ is the dominant singularity of the generating function $C_{\boldsymbol{\pi}}(z)$,*

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

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], a similar asymptotic estimate $-\pi_t n \frac{\frac{\partial}{\partial \pi_t}\rho(\boldsymbol{\pi})}{\rho}$ holds for $\mathbb{E}_n(N_t)$ and the weights computed by our methods can therefore be used by the recursive approach.

Input: Initial parameters z_0 and $\boldsymbol{\pi}$, 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 $\mathbb{E}_{z_0}(\boldsymbol{\pi})$;
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.

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 rational language with respect to the $\boldsymbol{\pi}$-weighted distribution.

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 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 [6], which we briefly present.

Classic Boltzmann sampling [6] By contrast with the recursive method, Boltzmann sampling, recently introduced by Duchon *et al* [6] 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. The probabilities associated with the different alternatives during the generation are computed by an external procedure called **oracle** (Symbolic algebra, or numerical method [10], 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 $\varepsilon > 0$.

Weighted Boltzmann generation Unlabelled Boltzmann samplers introduced by Flajolet *et al* [7] can be modified to draw objects from a weighted Boltzmann distribution (See Eq. 1) through a minor modification of the oracle. Indeed, one only needs to replace any occurrence of a 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, made explicit by Algorithm 2.

Input: the parameters $x, \boldsymbol{\pi}$
Output: An 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 $\Gamma_2\mathcal{A}(x, \boldsymbol{\pi}; n, \varepsilon)$

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

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

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 the rejection-based approach.

Complexity analysis Let us discuss the complexity of the exact-size and approximate-size rejection processes in the presence of weights. Our results relies on the analysis of flat distributions, found in [6] for uniform Boltzmann distributions.

Theorem 1 (Essentially proven in [6]). *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*
a) The approximate-size sampler $\Gamma_2\mathcal{C}(x_n, \boldsymbol{\pi}; n, \varepsilon)$ has expected running time bounded by $\frac{\kappa n}{\zeta_{\alpha_{\boldsymbol{\pi}}}(\varepsilon)} + c(\boldsymbol{\pi})$.
b) The exact-size sampler $\Gamma_2\mathcal{C}(x_n, \boldsymbol{\pi}; n, 0)$ has expected running time bounded by $\frac{\kappa\Gamma(\alpha_{\boldsymbol{\pi}})n^2}{\alpha_{\boldsymbol{\pi}}^{\alpha_{\boldsymbol{\pi}}}} + c(\boldsymbol{\pi})n$.
where κ is the cost-per-letter introduced by the canonical Boltzmann samplers, $\alpha_{\boldsymbol{\pi}}$ is the multiplicity of the dominant pole of $C_{\boldsymbol{\pi}}(z)$, $\zeta_{\alpha_{\boldsymbol{\pi}}}(\varepsilon) = \frac{\alpha_{\boldsymbol{\pi}}^{\alpha_{\boldsymbol{\pi}}}}{\Gamma(\alpha_{\boldsymbol{\pi}})}\int_{-\varepsilon}^{\varepsilon}(1+s)^{\alpha_{\boldsymbol{\pi}}-1}e^{-\alpha_{\boldsymbol{\pi}}(1+s)}ds$, $\Gamma(x)$ 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 it to be very limited and conjecture that the complexities observed for fixed-weights hold when targeted compositions are meaningful (Targeted occurrences of any letter in $[1, n-1]$).

Nevertheless, the following theorem provides a computable evaluation for the efficiency of the size-rejection. It is relied 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 $\mathbb{E}_{x,\boldsymbol{\pi}}(N) = n$, the approximate-size sampler $\Gamma_2\mathcal{C}(x_n, \boldsymbol{\pi}; n, \varepsilon)$ succeeds in the following mean number of trials :*

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

.

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. Namely, one draws objects in 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 $\Gamma_3\mathcal{A}(x, \boldsymbol{\pi}; n, \boldsymbol{m}, \varepsilon)$ 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 $x, \boldsymbol{\pi}, n, \boldsymbol{m}, \varepsilon$
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
$I(m_i(s), \varepsilon, \alpha) := [m_i(s) - m_i(s)^\alpha \varepsilon, m_i(s) + m_i(s)^\alpha \varepsilon]$
repeat
  $\gamma := \Gamma_2\mathcal{A}(x, \boldsymbol{\pi}; n, \varepsilon)$
until $\forall i, |\gamma|_i \in I(m_i(s), \varepsilon, \alpha)$;
return (γ)

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

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 . This situation allows for an analysis adapted from the bumpy case for the size of the parameters. Our main result is that, for any rational language whose automaton is strongly connected, a suitably parameterized multidimensional rejection phase will accept a word of target composition after $\mathcal{O}(n^{k/2})$ attempts. Moreover, allowing a $\mathcal{O}(\sqrt{n})$ tolerance on the number of occurrences of each letters, one obtains a sampler that succeeds in a number of attempts that is independent of n.

Now, let us denote by $U_n(\boldsymbol{\pi}_0)$ the k-multivariate random variable which follows the probability $\mathbb{P}(U_n(\boldsymbol{\pi}_0) = \boldsymbol{a}) = \frac{[z^n \boldsymbol{\pi}^{\boldsymbol{a}}] C_{\boldsymbol{\pi}}(z) \boldsymbol{\pi}_0^{\boldsymbol{a}}}{[z^n] C_{\boldsymbol{\pi}_0}(z)}$. That is to say the distribution of the parameters for the objects of the size n. Moreover, we denote by $\boldsymbol{\mu}(n, \boldsymbol{\pi}_0)$ is the mean-vector of $U_n(\boldsymbol{\pi}_0)$ and by $\boldsymbol{V}(n, \boldsymbol{\pi}_0)$ is its variance-covariance matrix. If we do not have any correlation between the parameters, the matrix $\boldsymbol{V}(n, \boldsymbol{\pi}_0)$ is positive definite (and so, invertible). So, we can 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 $\kappa(\boldsymbol{V}) := \inf_{||\boldsymbol{u}||_\infty = 1} \{||\boldsymbol{u}||_{\boldsymbol{V}}\}$[4], the infinum distance from the unit sphere to the center of the Banach space.

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

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

Theorem 3. *Let x_n and $\boldsymbol{\pi}_{\boldsymbol{a}}$ 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 $\boldsymbol{m}(s) = \mathbb{E}_{s,\boldsymbol{\pi}_{\boldsymbol{a}}}(N_i)$ and assume that :*
i) the standardized version $(U_n(\boldsymbol{\pi}_{\boldsymbol{a}}) - \boldsymbol{\mu}(n, \boldsymbol{\pi}_{\boldsymbol{a}}))/\mathfrak{f}(n)$ admits a limiting distribution when n tends to the infinity,
ii) the σ-concentrated condition is verified for $\sigma \leq 1$.
Then the rejection sampler $\Gamma_3\mathcal{C}(x_n, \boldsymbol{\pi}_{\boldsymbol{a}}; n, \boldsymbol{m}, \varepsilon)$ succeeds, in average, in least than

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

trials (of $\Gamma_2\mathcal{A}(x_n, \boldsymbol{\pi}; n, \varepsilon)$). This mean value tends to a constant as $n \to \infty$.
In particular, if the first rejection phase ($\Gamma_2\mathcal{A}(x_n, \boldsymbol{\pi}; n, \varepsilon)$) also achieves in average in constant time, then the overall cost of approximate size and composition sampling is $\mathcal{O}(n)$ on average.

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. *Assume that $(U_n(\boldsymbol{\pi}_{\boldsymbol{a}}))$ 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 rejection sampler $\Gamma_3\mathcal{C}(x_n, \boldsymbol{\pi}_{\boldsymbol{a}}; n, \boldsymbol{m}, 0)$ (exact composition) succeeds, in average, in $(2\pi)^{k/2}(\det(\boldsymbol{V}))^{1/2} = \mathcal{O}(f(n)^{k/2})$.*

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

In particular, assuming that the first rejection phase ($\Gamma_2\mathcal{A}(x_n, \boldsymbol{\pi}; n, \varepsilon)$) achieves in average in constant time, then the overall cost of approximate-size and exact composition sampling is in $\mathcal{O}(nf(n)^{k/2})$ on average.

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 On the Bender-Richmond-Williamson theorem

We prove in this section that the theorem applies to a large class of rational languages. Also, with only minor modifications, the ideas presented here can be extended to the more general transfer matrix models. From a combinatorial point of view, the most important properties of a rational language are **irreducibility** – strong-connectedness of the automaton – and **aperiodicity** – when words of any size exist. Let us first consider that the transition matrix $\boldsymbol{M}$ associated with the language is irreducible and aperiodic.

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. This condition guarantees that there is no trivial dependency between the parameters. 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, by 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 matrix proportional to n. In particular, for all $\sigma > 1/2$, the σ-concentrated condition holds.

Now, 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 we can also apply the theorem under the same assumptions (full dimensional periodicity parameter lattice and compact set with constant singular exponent). The question of the irreducibility is much more challenging and required serious developments that cannot be included in this short paper.

6 Sampling perfect Tetris histories

Let us first remind that the Tetris game consists in placing falling tetraminoes (or **pieces**) $\mathcal{P}$ one at a time in a $w \times h$ board. The goal of the player is to create hole-free horizontal lines which are subsequently eliminated, and the game continues until the pieces stack past the ceiling of the board.

Our focus in this application differs slightly from previous work dedicated to the complexity aspects of the game. Indeed, we are interested in the general **Tetris histories**, which consists in diagrams of pieces on the board after a game has been played, therefore disregarding the elimination of full lines. For the pieces to remain connected in such a diagram, we additionally assume that the lower blocks of a piece are always *consumed* before or simultaneously as its upper blocks. Under this hypothesis **perfect Tetris histories**, defined as histories of games ending with an empty board, are in bijection with tesselations of a $w \times h$ rectangle using tetraminoes. Our goal here is to generate such tesselations uniformly at random of fixed width w while constraining the number of occurrences of each piece to be uniform, capturing the *bag strategy* used by many implementations of the Tetris game.

6.1 Building the automaton of perfect histories

First let us find an unambiguous decomposition of Tetris histories. The idea is to focus on the state of the upper band of the tesselation that is not entirely filled, or **boundary** of an history. In particular

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
  $(Q, \sigma) \leftarrow (\mathcal{B}_w, \varnothing)$
  $S \leftarrow \{\mathcal{B}_w\}$
  while $S \neq \varnothing$ **do**
    $S \Rightarrow_{\text{pop}} \mathcal{B}$;
    for $p \in \mathcal{P}_{\mathcal{B}}$ **do**
      $\mathcal{B}' \leftarrow \mathcal{B} - p$;
      if $\mathcal{B}' \notin Q$ **then**
        $Q \leftarrow Q \cup \{\mathcal{B}'\}$;
        $S \Leftarrow_{\text{push}} \mathcal{B}'$;
      end
      $\sigma \leftarrow \sigma \cup \{(\mathcal{B}, p, \mathcal{B}')\}$;
    end
  end
  return (Q, σ)
end

Algorithm 4: Constructing the automaton $\mathcal{A}_w$ for tesselations of width w.

for tesselations 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 perfect histories.*

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 tesselations 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 tesselation has a upper-rightmost position which, upon removal, gives another tesselation of smaller size, and completeness of the decomposition propagates from tesselations 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 perfect histories 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 required for increasing values of w (See Table 2).

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

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	–

Table 2. Number of states in generated and minimal equivalent automata for various width.

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 of pieces of $n = 105$, one obtains the number of perfect histories

$$h_{6,105} = 336397650518572335440738353688656902745059634630110167876935128082282911.$$

One can then take suitable derivatives (See Equation 2) and extract the average composition in the uniform (unweighted) distribution, obtaining:

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

Consequently, the average composition of a perfect history 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 4.

6.3 From random perfect histories to Tetris instances

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

Proof. Let us assume that $\mathcal{T}$ is a perfect tesselation 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 and  , such that B exposes a southward wall 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.

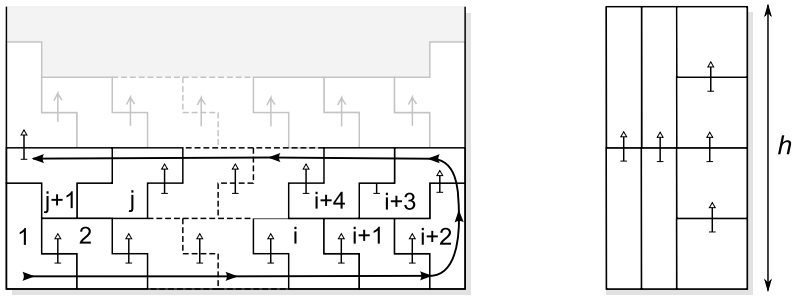

Fig. 2. Left: Perfect tetris histories associated with a unique instance. Only the most relevant dependency points are displayed here (arrows) and pieces are labelled with their rank in the only compatible instance. Duplicating the gadget preserves the uniqueness of the associated instance allowing for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

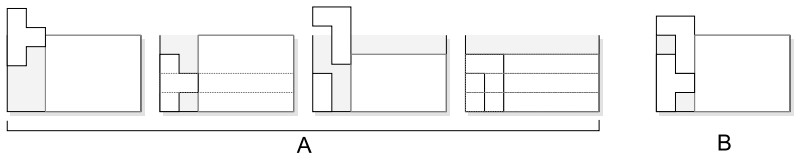

Fig. 3. Exemple of game ending with an empty board (Left) is not captured by our tesselation model (Right).

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. Indeed, a game can be successful in clearing the board while leaving temporary holes, as illustrated by Figure 3. 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 generate hard instances with the guarantee of a possible success (From the tesselation) which could be used to benchmark the performances of heuristics.

7 Conclusion

In this paper, we have presented and applied a general methodology for the multivariate random generation of combinatorial objects. Under natural hypotheses, random generators having complexity in $\mathcal{O}(n^{2+k/2})$ have been derived, outperforming the best algorithms so far in $\mathcal{O}(n^k)$. Furthermore, provided a small (linear) tolerance is allowed on the size of generated objects, and a $\Theta(\sqrt{n})$ one is allowed in the other dimensions, our generators generate objects in linear time. We have applied these principles to the generation of perfect Tetris histories, which we have generated and drawn.

On the Tetris side, it could be interesting to push further the generation of (possibly) hard Tetris instances using our polynomial-time constrained generators.

Also, this paper is the first step toward a general analysis of the multi-parameter Boltzmann sampling. Nevertheless, a lot of questions are still opened, in particular, about the difficult situation of non irreducible languages. Also the nature of the dependency between weights and frequencies is of uttermost interest, since it would allow us to address the complexity of our samplers in a more

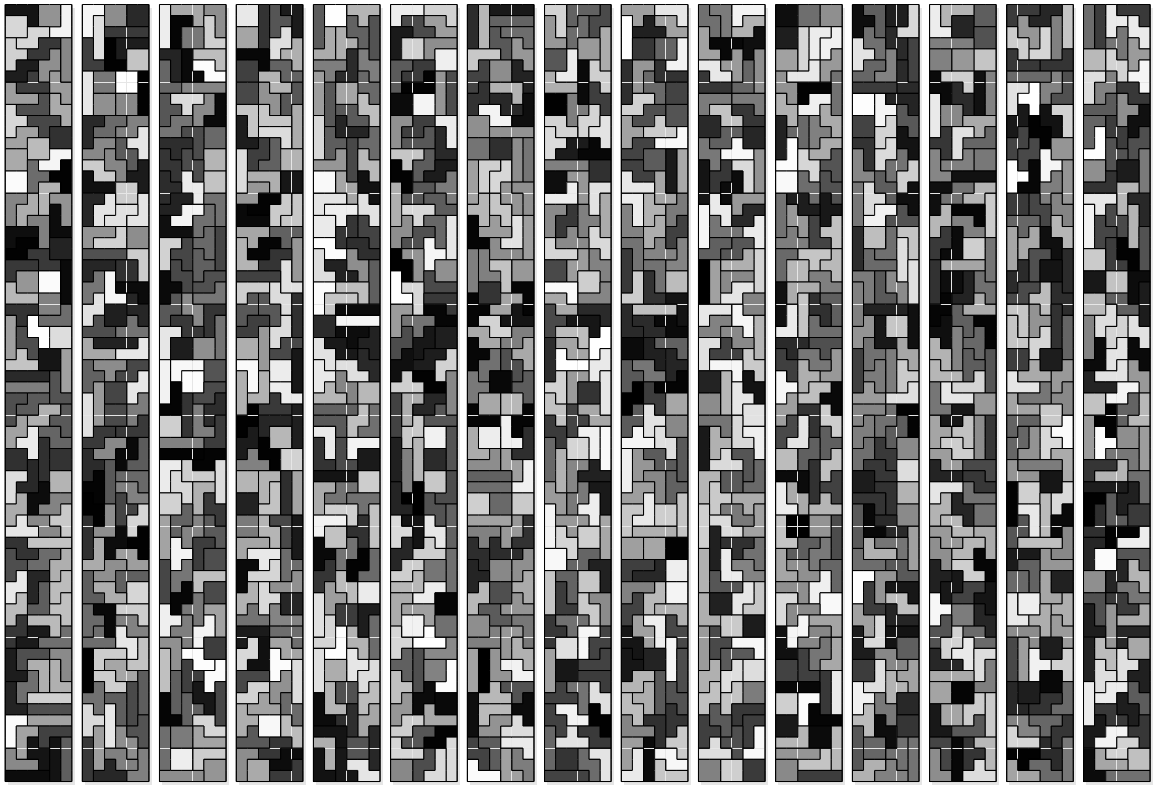

Fig. 4. Fifteen Tetris tesselations of width 6 having even composition (+/- 1) in the different pieces.

human-readable – yet rigorous – way. In a forthcoming work, we expect to tackle the analysis of multi-parameter Boltzmann samplers for the class of context-free languages. To conclude, we are convinced that multi-parameter Boltzmann samplers could be applied in a huge and various number of cases, such as constrained trees, permutations with a fixed number of cycles, functional graphs with a controlled number of components and so on. To some of us, that is FUN...

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. iii. 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. 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.
7. 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.
8. 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.
9. Hendrik Jan Hoogeboom and Walter A. Kosters, *Tetris and decidability*, Inf. Process. Lett. **89** (2004), no. 6, 267–272.
10. 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.
11. 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.
12. 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[5] 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* [10], 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 [10] 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 $\pi_c = J(\mathbb{E}_{z_0})^{-1}(\boldsymbol{\pi}) \cdot (n\boldsymbol{f} -$

[5] 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

$\mathbb{E}_{z_0}(\boldsymbol{\pi}))+\boldsymbol{\pi}$. 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.

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.

Proof (Proposition 3.). Immediate from the proof of [7], 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.