Fundamenta Informaticae 117 (2012) 85–101
DOI 10.3233/FI-2012-689
IOS Press

Boys-and-girls Birthdays and Hadamard Products

Olivier Bodini[*]
LIPN, Paris-XIII and UMR 7030, France
ANR project MAGNUM[†]
Olivier.Bodini@lip6.fr

Danièle Gardy
PRiSM, Université Versailles St Quentin
CNRS UMR 8144, France
gardy@prism.uvsq.fr

Olivier Roussel
LIP6, Paris-VI and UMR 7606, France
ANR project MAGNUM[†]
olivier.roussel@lip6.fr

Abstract. Boltzmann models from statistical physics, combined with methods from analytic combinatorics, give rise to efficient and easy-to-write algorithms for the random generation of combinatorial objects. This paper proposes to extend Boltzmann generators to a new field of applications by uniformly sampling a *Hadamard product*.

Under an abstract real-arithmetic computation model, our algorithm achieves approximate-size sampling in expected time $\mathcal{O}(n\sqrt{n})$ or $\mathcal{O}(n\,\sigma)$ depending on the objects considered, with σ the standard deviation of smallest order for the component object sizes. This makes it possible to generate random objects of large size on a standard computer. The analysis heavily relies on a variant of the so-called *birthday paradox*, which can be modelled as an occupancy urn problem.

[*] Address for correspondence: LIPN, Paris-XIII and UMR 7030 (France)

[†] ANR 2010 BLAN 0204

1. Presentation

Consider the classical birthday paradox: people arrive one by one; what is the expected time we have to wait until two people have a common birthday? The answer is surprisingly low: 25. Now assume that we only consider a birthday if it involves a boy and a girl: what is the expected time? The answer is 34. The classical birthday paradigm has appeared at various times in the modelling and analysis of algorithms, but such a (rather natural) extension has received, until now, limited attention. We present in this paper an algorithm for the random generation of Hadamard products, whose analysis relies on a boys-and-girls birthday model.

In 2004, Duchon, Flajolet, Louchard and Schaeffer [4] proposed a new model, the so-called Boltzmann model, which leads to the systematic construction of samplers for random objects in combinatorial classes described by specification systems. This framework has two main features: *uniformity* — i.e. two objects of the same size have equal chances of being drawn — and *quasi-linear complexity*,which makes possible the efficient generation of huge objects, to address problems of testing and benchmarking.

Boltzmann samplers depend on a real parameter, and generate an object with a probability that depends only on its size. Actually the size of the sampler output follows a Boltzmann probability distribution: the probability that a random object has size n is proportional to x^n for some parameter x, which can be tuned to achieve a chosen average size. Moreover, using rejection, one can obtain efficient exact size or approximate size samplers. This approach differs from the ”recursive method” introduced by Nijenhuis and Wilf [13, 9], in that it gives the possibility of relaxing the constraint of exact-size output. Since no preprocessing phase is needed, this implies a significant gain in complexity and approximate-size sampling can be done in expected linear time in a variety of cases. Boltzmann samplers have been developed for a whole set of combinatorial classes: labelled, unlabelled, and colored [4, 6, 3]. Such classes are defined from basic elements by means of fundamental constructions, well known in combinatorics [8]. The present paper is part of this joint effort to obtain Boltzmann generators for all usual classes and constructors, and focuses on the construction of an approximate-size sampler for the Hadamard product of two combinatorial classes.

In many cases, it is useful to have two objects of the same size, in order to visualize some bias on the properties, e.g., to evaluate the typical height of a tree, or the number of components in a composed structure. Hadamard products also appear naturally when building standard combinatorial objects such as a drunkard’s walk or a partition of graphs into cycles; see also [2] for a recent application to automata.

In other words, we study sampling for combinatorial objects defined as pairs of equal-size components, and extend the basic random sampling model to generate efficiently such objects. The main idea is to use a (suitably tuned) rejection method, whose efficiency can be proved by an argument extending the classical birthday problem for the waiting time of the first collision, to an urn model with colored balls.

The plan of the paper is as follows. We recall the notion of Boltzmann sampler in the next section, then describe in Section 3 an approximate-size sampler for the Hadamard product of two classes, assuming we already know an approximate-size sampler (for instance a Boltzmann sampler) for each of those classes. Section 4 is devoted to an example of Boltzmann sampling for the Hadamard product: we apply our method to generate huge constrained random walks in the plane. We turn back to the detailed analysis of the complexity in Section 5, where we prove that the limit distribution of the first collision in a generalized urn model follows a Rayleigh distribution, which in turn gives the expected complexity of the algorithm. As a consequence, the time complexity of our approximate-size sampler is

$\mathcal{O}(n\sqrt{n})$ under reasonable assumptions. Finally, we consider the limits and possible extensions of our approach in Section 6.

2. Boltzmann samplers and combinatorial structures

2.1. Combinatorial classes

Definition 2.1. A *combinatorial class* $\mathcal{C}$ is a countable (or finite) set, with a *size* function $|\cdot| : \mathcal{C} \mapsto \mathbb{N}$ and such that there are only finitely many objects of each size.

Each class has an *ordinary generating function* defined by

$$C(z) = \sum_{\gamma\in\mathcal{C}} z^{|\gamma|} = \sum_{n\in\mathbb{N}} c_n z^n.$$

We use the following notations: let $\mathcal{C}$ be a class, and γ be any object in $\mathcal{C}$; then its size is $|\gamma|$. Furthermore $\mathcal{C}_n = \{\gamma \in \mathcal{C} \mid |\gamma| = n\}$ and $c_n = \mathrm{Card}(\mathcal{C}_n)$.

Decomposable classes can be constructed from basic objects, called *atoms*, and from a set of operators — such as cartesian product, sequence, set or cycle — allowing us to build large objects out of smaller ones. In order to construct composed objects from the basic ones, we need a set of rules that allow us to build an object from simpler ones. A few of these operators are presented on Figure 1; see also [6, 4, 8] for a more extensive list.

$\mathcal{A}$	Description	$A(z)$	$\Gamma\mathcal{A}(x)$
ε	Empty class	1	**return** ε
$\mathcal{Z}$	Atomic class	z	**return** $\mathcal{Z}$
$\mathcal{B}\times\mathcal{C}$	Cartesian product	$B(z)\times C(z)$	**return** $(\Gamma\mathcal{B}(x), \Gamma\mathcal{C}(x))$

$\mathcal{B}+\mathcal{C}$	Disjoint union	$B(z)+C(z)$	**if** *Bernoulli*$\left(\frac{B(x)}{B(x)+C(x)}\right)$
			then return $\Gamma\mathcal{B}(x)$
			else return $\Gamma\mathcal{C}(x)$

$\mathcal{S}eq(\mathcal{B})$	Sequence	$\frac{1}{1-B(z)}$	$l :=$ *Geometric*$(B(x))$ **return** $\underbrace{(\Gamma\mathcal{B}(x),\dots,\Gamma\mathcal{B}(x))}_{l\text{ times}}$

Figure 1. Some classical constructors, with their ordinary generating function and Bolzmann sampler

2.2. Boltzmann samplers

The Boltzmann model is a simple and generic framework to sample efficiently combinatorial objects. It ensures that each object of a given size has the same probability to be drawn. A *Boltzmann sampler* $\Gamma\mathcal{C}(x)$ for an (unlabelled) combinatorial class $\mathcal{C}$ is a random generator that produces objects of $\mathcal{C}$, in such a way that the probability of drawing a given object $\gamma \in \mathcal{C}$ of size n is exactly

$$\mathbb{P}_x(\gamma) = \frac{1}{C(x)} x^{|\gamma|} = \frac{1}{C(x)} x^n.$$

Since the probability for any object γ depends only on its size, not on its shape, the probability density induced on the objects of a fixed size is uniform.

Such a sampler comes in two flavors: the *free* Boltzmann sampler depends on a parameter x, and generates an object of expected size $\mathbb{E}_x(\text{size of the output}) = xC'(x)/C(x)$; the *approximate-size* Boltzmann sampler starts from a free sampler, followed by a rejection step to ensure that the output has size in $[(1-\epsilon)n, (1+\epsilon)n]$. Moreover, one can build automatically a sampler according to the specification of a combinatorial class, by following recursively the rules described in [4].

We can classify combinatorial classes according to the generic shape of the probability distribution of the size of a random object: it is either flat, bumpy, or peaked [4]. As the peaked case can be transformed into a flat distribution by pointing, we shall not consider it.

The precise definitions of these distributions are given in the section 5.3; here we shall use the following results (see again [4] for the proof).

Fact 2.2. Let $\mathcal{C}$ be a combinatorial class with Boltzmann variance σ_C^2; the time for Boltzmann generation of a random sample in $\mathcal{C}$ in approximate size is $\mathcal{O}(n)$. In exact size, it is $\mathcal{O}(n\,\sigma_C)$ if the distribution of $\mathcal{C}$ is bumpy and $\mathcal{O}(n^2)$ if it is flat.

2.3. The Hadamard product

We next introduce the central object of this paper: the *Hadamard product*, for which we extend the Boltzmann formalism. The notion of the Hadamard product is a fairly old one; it appeared in J. S. Hadamard's 1899 paper *Théorème sur les séries entières* [11].

Definition 2.3. The *Hadamard product* of two classes $\mathcal{B}$ and $\mathcal{C}$, denoted by $\mathcal{B} \odot \mathcal{C}$, is the subset of $\mathcal{B} \times \mathcal{C}$ such that the two objects in a pair have exactly the same size. Furthermore, if $\alpha = (\beta, \gamma) \in \mathcal{A}$, we define the size of α as

$$|\alpha| = |\beta| = |\gamma|.$$

The generating function of $\mathcal{A} = \mathcal{B} \odot \mathcal{C}$ is

$$A(z) = \sum_{n=0}^{\infty} a_n z^n = \sum_{n=0}^{\infty} b_n c_n z^n = \frac{1}{2i\pi} \oint \frac{B(\xi)}{\xi} C\left(\frac{z}{\xi}\right) \mathrm{d}\xi.$$

where the contour is a circle around the origin taken inside the domain of analyticity of both B and C. It is the entrywise product of the two generating functions $B(z)$ and $C(z)$.

3. An approximate-size sampler for the Hadamard product

Boltzmann samplers have been built for a variety of combinatorial operators (union, cartesian product, cycle, ...); our aim in this paper is to present and analyze such a sampler for the Hadamard product

$\mathcal{A} = \mathcal{B} \odot \mathcal{C}$. There is a significant difference with Boltzmann samplers for the (classical) Cartesian product [4, 8, 6]: there, only one drawing is involved for each component of the pair under construction; here we almost always need to draw more than one object for each component.

From now on, and for the sake of simplicity, we will say that ”a class $\mathcal{A}$ is flat” rather than the more rigorous and verbose ”the probability distribution of the size of a random object for the Boltzmann generator of the class $\mathcal{A}$ is flat”, and analogously for bumpy.

We consider now how we can obtain an approximate-size sampler for the Hadamard product $\mathcal{A} = \mathcal{B} \odot \mathcal{C}$. We present two algorithms: the first one samples first $\mathcal{B}$ in approximate size, then $\mathcal{C}$ in exact size; the second one draws at each step samples of both $\mathcal{B}$ and $\mathcal{C}$, and requires that we keep the sets of objects obtained until this point. A variant of this algorithm decides randomly at each step to draw an instance either of $\mathcal{B}$ or of $\mathcal{C}$. We give below the first algorithm, whose principle is obvious.

```
Algorithm 1: Naïve approximate-size sampler $\Gamma\mathcal{A}$ for $\mathcal{A} = \mathcal{B} \odot \mathcal{C}$
  Input: The expected range $R = [(1-\varepsilon)n, (1+\varepsilon)n]$ for the output
  Output: An object of $\mathcal{A}$ with size in $R$
  Draw a instance of $\mathcal{B}$ in approximate size.;
  Draw an instance of $\mathcal{C}$ with exact size, the size of the instance of $\mathcal{B}$ obtained in the first step.
```

Using Fact 2.2, we obtain at once the behaviour of this algorithm:

Theorem 3.1. The expected time required to generate in approximate size a Hadamard product of size n from Algorithm 1 is asymptotically

- $\mathcal{O}(n^2)$ when $\mathcal{B}$ and $\mathcal{C}$ both follow a flat distribution,
- $\mathcal{O}(n^2)$ when $\mathcal{B}$ is bumpy and $\mathcal{C}$ is flat,
- $\mathcal{O}(n\sigma_C)$ when $\mathcal{B}$ is flat and $\mathcal{C}$ is bumpy with variance σ_C^2,
- $\mathcal{O}(n\sigma)$ when both distributions are bumpy and $\sigma = \min(\sigma_B, \sigma_C)$.

Proof:
When $\mathcal{C}$ is flat, we draw an approximate-size sample for $\mathcal{B}$ in time $\mathcal{O}(n)$ when $\mathcal{B}$ is flat or bumpy, then an exact-size sample for $\mathcal{C}$ in time $\mathcal{O}(n^2)$. When $\mathcal{B}$ follows a flat distribution and $\mathcal{C}$ a bumpy one, we draw an approximate-size sample for $\mathcal{B}$ in time $\mathcal{O}(n)$, then an exact-size sample for $\mathcal{C}$ in time $\mathcal{O}(n\,\sigma_C)$. If $\mathcal{B}$ and $\mathcal{C}$ both follow a bumpy distribution, assume that $\sigma_C = \mathcal{O}(\sigma_B)$; again we draw an object of $\mathcal{B}$ in approximate-size in time $\mathcal{O}(n)$ and an exact-size object of $\mathcal{C}$ in time $\mathcal{O}(n\,\sigma_C)$. □

The idea underlying the second algorithms is as follows. As the Hadamard product builds pairs from independent components, we keep a set of potential candidates for each of the components, until we find in these sets an element of $\mathcal{A}$ and an element of $\mathcal{B}$ of the same size. This is basically a tuned rejection sampler: we draw objects from $\mathcal{B}$ and $\mathcal{C}$, until we have an object of $\mathcal{B}$ and an object of $\mathcal{C}$ (usually not drawn at the same time) with the same size. Note that this algorithm is an uniform sampler for $\mathcal{B} \odot \mathcal{C}$, but it is not a Boltzmann sampler.

Algorithm 2: Approximate-size sampler $\Gamma\mathcal{A}$ for $\mathcal{A} = \mathcal{B} \odot \mathcal{C}$

```
Input: The expected range $R = [(1-\varepsilon)n, (1+\varepsilon)n]$ for the output; the positive probabilities $q_1$
       and $q_2$
Output: An object of $\mathcal{A}$ with size in $R$
B ← ∅ ; C ← ∅ ;
Choose $x_1$ such that the expected size of the output of $\Gamma\mathcal{B}(x_1)$ is $n$ ;
Choose $x_2$ such that the expected size of the output of $\Gamma\mathcal{C}(x_2)$ is $n$ ;
repeat
    Decide to draw a sample of $\mathcal{B}$ with probability $q_1$, or of $\mathcal{C}$ with probability $q_2$ ;
    if $\mathcal{B}$ is chosen then
        Randomly draw an object $b$ from the class $\mathcal{B}$ using $\Gamma\mathcal{B}(x_1)$ of size in $R$ ;
        if size of object is new then $\mathrm{B} \leftarrow \mathrm{B} \cup \{b\}$
    else
        Randomly draw an object $c$ from the class $\mathcal{C}$ using $\Gamma\mathcal{C}(x_2)$ of size in $R$ ;
        if size of object is new then $\mathrm{C} \leftarrow \mathrm{C} \cup \{c\}$
until $\exists (b, c) \in \mathrm{B} \times \mathrm{C}, |b| = |c|$ ;
return $a = (b, c)$;
```

Our algorithm generates *sets* of objects for both $\Gamma\mathcal{A}(x)$ and $\Gamma\mathcal{B}(x)$. Its execution time will be closely related to the cardinalities of those sets, i.e. to the number of elements drawn. As we build sets from $\mathcal{A}$ and $\mathcal{B}$, the space complexity might also be in question. The key to analyze these complexities is a modelization as a generalized birthday problem, which we present in Section 5, where we prove the following result.

Theorem 3.2. The expected time required to generate in approximate size a Hadamard product of size n from Algorithm 2 is asymptotically

- $\mathcal{O}(n\sqrt{n})$ if both $\mathcal{B}$ and $\mathcal{C}$ follow a flat distribution;
- $\mathcal{O}(n\,\sigma_B)$ if $\mathcal{B}$ follows a bumpy distribution and $\mathcal{C}$ a flat one;
- $\mathcal{O}(n\,\sigma)$ with $\sigma = \min(\sigma_B, \sigma_C)$ when both $\mathcal{B}$ and $\mathcal{C}$ are bumpy.

The expected space is of the same order as the expected time.

Note that we can omit the case $\mathcal{B}$ flat and $\mathcal{C}$ bumpy, as it is — *mutatis mutandis* — similar to the second case $\mathcal{B}$ bumpy and $\mathcal{C}$ flat. Indeed, Algorithm 2 is similar for $\mathcal{B} \odot \mathcal{C}$ and $\mathcal{C} \odot \mathcal{B}$ up to the order of the output pair.

Theorems 3.1 and 3.2 together show that **Algorithm 2 outperforms Algorithm 1**.

The actual choice of the probabilities q_1 and $q_2 = 1 - q_1$ does not change the results of this paper (as long as they are both positive); for example we can assume $q_1 = \frac{1}{2} = q_2$. However, we do not have a precise view about how to choose them wisely, and did not devised a way to choose them depending on the classes $\mathcal{B}$ and $\mathcal{C}$. Nonetheless, it is easy to see that the choice $q_1 = q_2$ is not always optimal. For example, if $\mathcal{B}$ is bumpy and $\mathcal{C}$ flat, it is a good choice to have $q_2 > \frac{1}{2} > q_1$: as sizes of objects drawn from $\Gamma\mathcal{C}$ will spread more that those from $\Gamma\mathcal{B}$, we have to draw more of them until a collision occurs.

Furthermore, if we want to produce a large number of objects of similar sizes from the similar specification, a situation which we often encounter in practice, we can think about a variant of this algorithm. From one call to another with the same parameter, we could keep in memory the sets B and C so that a collision would occur faster. And when such a collision occurs, we simply remove the corresponding objects from B and C.

4. Example: A very drunk man vs. the classical drunkard

We give in this section an example where our approximate size Hadamard sampler is used to build a random object of large size. The problem we propose to examine is the following one, derived from the classical drunkard walk.

The drunkard's walk starts from the origin, does a zigzag walk without memory — this is a Markov process — and goes back to the origin. More precisely, consider here a walk on the first quadrant of the plane where each move corresponds to a translation following the vector $NE = (1, 1)$, $SE = (1, -1)$, $NW = (-1, 1)$ or $SW = (-1, -1)$. The projection of a walk of n steps on the axis (Ox) (resp. (Oy)) is a Dyck path of length n. Conversely, choose two Dyck paths of the same length on $\{x, \bar{x}\}$ and $\{y, \bar{y}\}$: the drunkard's walk is obtained by choosing, at step i, NE for xy, SE for $x\bar{y}$, NW for $\bar{x}y$ or SW for $\bar{x}\bar{y}$. As is well known [1, 16], it is possible to draw a Dyck path of exact length n in linear time: for this simple example, our algorithm is not needed and the time for drawing a drunkard's walk is linear.

Now assume that the man is more than usually drunk, and that he does not make three consecutive moves in any direction. In other words, the projection of his walk on the axis (Ox) (resp. the axis (Oy)) is a Dyck path without three consecutive identical steps, and the walk is the Hadamard product of two constrained paths. Such a path admits the following simple context-free specification : $D = 1 + xyD + xxyyD + xxyDxyyD$, whose generating function — when the size of a path is its number of *steps*, e.g. xy — is $d(z) = (1 - z - z^2 - \sqrt{1 - 2z - z^2 - 2z^3 + z^4})/2z^3$ with dominant singularity $\rho = (3 - \sqrt{5})/2 \simeq 0.381966012$. The constrained Dyck paths are of peaked type [4], and we generate a random path in linear time by pointing.

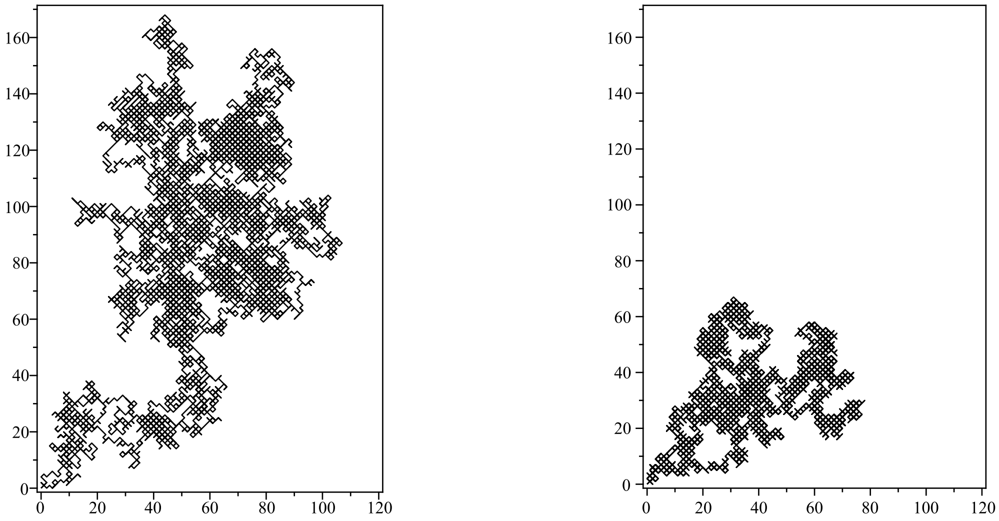

Figure 2. Left a classical drunkard walk; right a very drunk man walk with the same size

Should we want to compare this walk to the classical drunkard's walk, our sampler can generate very large such walks. Figure 2 shows two random walks. The constrained Dyck paths were generated first, with a parameter $x = 1 - \rho \simeq 0.61803398$, which gives an expected size $n = 14376$. We obtained a constrained walk of length $n = 14561$, then generated two random (classical) Dyck paths of the same length.

We conducted some experiments about the time needed to compute such very drunk man walks. We randomly created 2000 random walks using our algorithm — with the same value for x as before, aiming at a size of 14376 — on a desktop computer with a 2.66GHz processor. On average, with a naive implementation and without memory between each run, it took 43 seconds to generate a walk, with a standard deviation of 67 seconds. Half of the drawings have been finished in less than 20 seconds.

On this random sample, we can observe that the very drunk man explores a smaller and more compact part of the quarter plane. Possible parameters for the comparison might be the extremal positions of the walk, or the "area" of the walk, as measured for example by the smallest including rectangle, or by the convex hull.

5. A birthday problem, and complexity of Hadamard sampling

In this section we first present a modelisation of our algorithm as a birthday problem, then outline the proof of its complexity. A result of Selivanov [17] is central to our analysis ; this result has been recently improved for more distributions [10]. We next examine some special cases, before turning to distributions obtained from Boltzmann generators.

5.1. Urns models and the birthday paradox

The parameter that determines the performance of our algorithm is the time to obtain for the first time an object of $\mathcal{B}$ and an object of $\mathcal{C}$ of the same size. As we keep sets of all the objects of $\mathcal{B}$ and $\mathcal{C}$ we have drawn, the time of our algorithm will be the waiting time for the first collision, i.e. the first time we obtain *an object of $\mathcal{B}$ and an object of $\mathcal{C}$ of the same size*.

This can be interpreted as a variant of the classical birthday problem, as follows. Let n be the expected size of the objects drawn, and ϵ the tolerance parameter: we keep those objects of $\mathcal{B}$ or $\mathcal{C}$ which have size in the intervall $[n(1-\epsilon), n(1+\epsilon)]$. The number of sizes for which we keep an object, or equivalently the number of urns (each urn being labelled by a size) is $N = 2n\epsilon - 1$. Each object (ball) can have two colors (one for $\mathcal{B}$ and the other for $\mathcal{C}$). With this framework, Hadamard sampling amounts to the following random allocation problem:

- We choose a color $k = 1, 2$ for a ball with a constant probability q_k: $q_1 + q_2 = 1$.
- Knowing that the ball we draw has color k, we put it in the urn i with probability $p_{k,i}$: $p_{k,i}$ is the conditional probability of drawing urn i, assuming that the ball has color k, and $\sum_{i=1}^{N} p_{k,i} = 1$.
- The probability that a random ball goes to urn i is thus $p_i = p_{1,i}q_1 + p_{2,i}q_2$.

We define a *collision,* or a *birthday,* as *two balls of different colors in the same urn.* Now let τ_N be the random variable equal to the number of balls we have to draw until the first collision: τ_N gives precisely the number of trials before we obtain our Hadamard product.

The birthday problem, or paradox, is a well-known problem of discrete probability; we refer the reader to classical treatises on this subject, e.g., Feller [5], for the standard approach, and to [7] for its presentation in a combinatorial framework and generalization. The colored-balls problem is not a classical variation; to the best of our knowledge its first appearance was in 1968, when Popova [15] considered balls of two colors and non-uniform urns, and gave the joint limiting distribution for the numbers of urns of a single color and of the two colors. It then appeared in a paper by Nishimura and Subaya in the mid-eighties [14], where they considered uniform urns. The most significant result, for our point of view, came ten years later with Selivanov [17], who considered the general case of k colors, and gave first a Poisson approximation for the probability that there is no collision after n balls, under some conditions on the probability distributions for the colors and the urns, then proved a limiting theorem for the waiting time until the first collision, reframed here for two colors:

Theorem 5.1. (Selivanov [17])
Assume that $N \to \infty$ and that q_1 and q_2 are constant, with $w_2 = q_1^2 + q_2^2 = 1 - 2q_1q_2$. Assume furthermore that

- **(C1)** $v_2 := \sum_i p_i^2$ is $o(1)$ when $N \to +\infty$;
- **(C2)** $p_{\max} := \max_i\{p_i\}$ is such that $p_{\max}/\sqrt{v_2} < C$ where C is a constant (independent of N).

Define a normalization factor $\alpha = \sqrt{v_2(1-w_2)} = \sqrt{2v_2q_1q_2}$. Then the normalized variables $\alpha\tau_N$ are asymptotically distributed with Rayleigh distribution of density $te^{-t^2/2}1_{t\geq 0}$, and

$$\mathbb{E}[\tau_N] = \sqrt{\frac{p_{\max}}{2v_2(1-w_2)}}\,(1+o(1)) = \sqrt{\frac{p_{\max}}{4v_2q_1q_2}}\,(1+o(1)).$$

5.2. Some extremal cases

Selivanov's result applies for example when both probability distributions $(p_{1,i})$ and $(p_{2,i})$ on the urns are uniform: it is easy to check that $v_2 = 1/n$, which leads to $\mathbb{E}[\tau_N] = \sqrt{p_{\max}n/4q_1q_2}$. When the probability distributions are "not far from the uniform", in the sense that all $p_{k,i}$ ($k = 1, 2$) belong to some interval $[a_k/N, b_k/N]$ for constant a_k and b_k, again $v_2 = \alpha/n$ for some constant α which can be computed, and $\mathbb{E}[\tau_N] = \sqrt{p_{\max}n/4\alpha q_1q_2}$.

We assume next that the N urns have the same probability distribution: this appears e.g. when $\mathcal{B} = \mathcal{C}$, a case which is in some sense extremal for the analysis.

Lemma 5.2. Assume that the probability distributions $(p_{1,i})$ and $(p_{2,i})$, $1 \leq i \leq N$, are identical. Then the expected value of τ_N is maximal when $p_{1,n}$ and $p_{2,n}$ are uniformly distributed, and is equal to $\sqrt{p_{\max}n/4q_1q_2}$.

Proof:
Write p_i for the common value $p_{k,i}$, $k = 1...2$. If the distribution is not uniform, there exists some i, such that $p_i \neq p_{i+1}$. Define a probability distribution $(p'_{.})$ from $(p_{.})$: p'_j equal to p_j for $j \neq i, i+1$, $p'_i = \lambda p_i + (1-\lambda)p_{i+1}$, and $p'_{i+1} = (1-\lambda)p_i + \lambda p_{i+1}$. We compare the expected values of the first collision when the distributions are (p_i) and (p'_i). Let T' be the random variable for the collision time with the new distribution. A standard computation shows that $\frac{\partial}{\partial\lambda}\mathbb{E}(T') = 0$ when $\lambda = \frac{1}{2}$, and that this

is a maximum. For this new distribution, $p'_i = p'_{i+1}$. By compacity, it follows that the worst case is obtained when the common distribution p_i is uniform. □

Another extremal case appears when one of the objects, say $\mathcal{B}$, is uniformly distributed, and the other object, $\mathcal{C}$, follows a Dirac distribution. Then the waiting time for the first collision is the waiting time until a specified value is obtained under the uniform distribution, and follows a geometric distribution of parameter $1/N$; as a consequence the expected waiting time is linear in this situation.

Fact 5.3. Assume that the distribution $(p_{1,i})$ is uniform. Then the worst case is obtained when the distribution $(p_{2,i})$ is a Dirac distribution, i.e. when some $p_{2,k}$ is equal to 1.

5.3. Boltzmann distributions

Following [4], we consider combinatorial classes for which the probability distribution for the size of a random object is either flat or bumpy, – the peaked case being transformed, by pointing, into a flat distribution. Of course, the size of $\mathcal{C}$ can follow a similar behavior. We recall below the characterization of flat or bumpy distributions and give some general results on coefficients or values of the functions at the Boltzmann parameter, before turning to the analysis of our algorithm in subsequent sections.

Definition 5.4. Let N be the size of a random object drawn with parameter x; its first moments are $\mu_1(x) = \mathbb{E}_x(N)$ and $\mu_2(x) = \mathbb{E}_x(N^2)$; its variance is $\sigma^2(x) = \mu_2(x) - \mu_1(x)^2$. Let ρ be the dominant singularity of the generating function enumerating the objects.

- The Boltzmann distribution of a combinatorial class $\mathcal{C}$ is *flat* if its generating function $C(z)$ is analytic at 0 with a finite radius of analyticity $\rho > 0$ and satisfies the following two Δ-singular conditions:

  (i) The function $C(z)$ admits ρ as its only singularity on the circle $|z| = \rho$ and it is continuable in a domain $\Delta(r,\theta) = \{z|z = \rho, |z| < r, \arg(z-\rho) \in (-\theta,\theta)\}$, for some $r > \rho$ and some θ satisfying $0 < \theta < \pi/2$;

  (ii) For z tending to ρ in the Δ-domain, $C(z)$ satisfies a singular expansion of the form

  $$C(z) \sim_{z\to\rho^-} c_0(1 - z/\rho)^{-\alpha} + o((1 - z/\rho)^{-\alpha}), \alpha \in \mathbb{R}^+.$$

  The quantity $-\alpha$ is called the singular exponent of $C(z)$.

- The Boltzmann distribution of a combinatorial class $\mathcal{C}$ is *bumpy* [1] if $\sigma(x)/\mu_1(x) \to 0$ for $x \to \rho^-$, which amounts to $\mu_2(x)/\mu_1(x)^2 \to 1$, $\sigma(x) \to +\infty$ and $C(z)$ is H-admissible: there exists a function $\delta(x)$ defined for $x < \rho$ with $0 < \delta(x) < \pi$, such that for $|\theta| < \delta(x)$ as $x \to \rho^-$, $C(xe^{i\theta}) \sim C(x)e^{i\mu_1(x)\theta - 1/2\sigma^2(x)\theta^2}$. Uniformly as $x \to \rho^-$, for $\delta(x) \le |\theta| \le \pi$, one gets $C(xe^{i\theta}) = o\left(\frac{C(x)}{\sigma(x)}\right)$.

[1] The definition of a bumpy distribution given in [4] is restricted to the first condition; however the conditions we add are very natural; they hold for the bumpy distributions met in practice.

We next give some approximation results on the coefficients, that we shall need later on.

Lemma 5.5. Let $\mathcal{C}$ be a combinatorial class with generating function $C(z)$, and x_n be the solution of the equation $x_n C'(x_n)/C(x_n) = n$.

- If $\mathcal{C}$ follows a flat distribution, then $\tilde{x}_n = \rho(1 - \alpha/n)$ is an approximation of the Boltzmann parameter x_n and

$$\begin{aligned} c_m &\sim \frac{c_0}{\Gamma(\alpha)}\,\rho^{-m}\,m^{\alpha-1}; \\ C(x_n) &\sim c_0\left(\frac{n}{\alpha}\right)^{\alpha}; \\ \frac{c_m\,x_n^m}{C(x_n)} &\sim \frac{\alpha^{\alpha}e^{-\alpha}}{\Gamma(\alpha)}\cdot\left(\frac{m}{n}\right)^{\alpha}\cdot\frac{1}{m} = \Theta\left(\frac{1}{m}\right); \end{aligned} \tag{1}$$

  when $m \in [an, bn]$ for some constants a and b.

- If $\mathcal{C}$ follows a bumpy distribution, then $x_n \to \rho^-$ where ρ is the (finite or infinite) singularity of $C(z)$ and

$$\begin{aligned} c_m &\sim \frac{C(x_n)}{x_n^n \sigma_C(x_n)\sqrt{2\pi}}; \\ \frac{c_m\,x_n^m}{C(x_n)} &= \frac{1}{\sigma_C(x_n)\,\sqrt{2\pi}}\ \exp\left(-\frac{(m-n)^2}{2\,\sigma_C(x_n)^2} + o(1)\right); \end{aligned}$$

  uniformly for all m as $n \to +\infty$.

Proof:
When the distribution of $\mathcal{C}$ is flat, standard singularity analysis gives

$$c_m \sim \frac{c_0}{\Gamma(\alpha_c)}\,\rho_c^{-m}\,m^{\alpha_c-1}.$$

The approximation $\tilde{x}_n$ of x_n comes from the definition of a flat distribution; see [4], and

$$C(x_n) \sim c_0\left(\frac{n}{\alpha_c}\right)^{\alpha_c}.$$

Putting all this together, we obtain

$$\begin{aligned} \frac{c_m\,x_n^m}{C(x_n)} &\sim \frac{c_0}{\Gamma(\alpha_c)}\,\rho_c^{-m}\,m^{\alpha_c-1}\cdot(\rho_c\,(1-\alpha_c/n))^m\cdot\frac{1}{c_0}\left(\frac{n}{\alpha_c}\right)^{-\alpha_c} \\ &= \frac{\alpha_c^{\alpha_c}}{\Gamma(\alpha_c)}\cdot\frac{1}{m}\cdot\left(\frac{m}{n}\right)^{\alpha_c}\cdot(1-\alpha_c/n)^m. \end{aligned}$$

Assuming that $m/n \in [a, b]$ gives the result.
The case of a bumpy distribution comes from Hayman [12]; see also [4, p. 25]. □

5.4. Selivanov's theorem applied to Boltzmann distributions

Define $I_\epsilon = [(1-\epsilon)n, (1+\epsilon)n]$ and[2] $B_\epsilon(z) = \sum_{i \in I_\epsilon} b_i z^i$, $C_\epsilon(z) = \sum_{i \in I_\epsilon} c_i z^i$. The probability that we draw an object of $\mathcal{B}$ of size i, with $i \in I_\epsilon$, is $p_{1,i} = \ b_i x^i / B_\epsilon(x)$, where b_i is the number of objects of $\mathcal{B}$ with size i and $x := x_n$ is defined by the equation $xB^{'}(x)/B(x) = n$. A similar result holds for $\mathcal{C}$, with a parameter $y := y_n$. What are the conditions on $\mathcal{B}$ and $\mathcal{C}$ that allow us to use Selivanov's theorem?
The (unconditional) probability for a ball to fall into urn i is

$$p_i = q_1 p_{1,i} + q_2 p_{2,i} = q_1 \frac{b_i x^i}{B_\epsilon(x)} + q_2 \frac{c_i y^i}{C_\epsilon(y)},$$

and Selivanov's parameter $v_2 = \sum_i p_i^2$ is here

$$\begin{aligned} v_2 &= q_1^2 \frac{\sum_{i \in I_\epsilon} b_i^2 x^{2i}}{B_\epsilon(x)^2} + 2q_1 q_2 \frac{\sum_{i \in I_\epsilon} b_i c_i x^i y^i}{B_\epsilon(x) C_\epsilon(y)} + q_2^2 \frac{\sum_{i \in I_\epsilon} c_i^2 y^{2i}}{C_\epsilon(y)^2} \\ &= q_1^2 \frac{B_\epsilon \odot B_\epsilon(x^2)}{B_\epsilon(x)^2} + 2q_1 q_2 \frac{B_\epsilon \odot C_\epsilon(xy)}{B_\epsilon(x) C_\epsilon(y)} + q_2^2 \frac{C_\epsilon \odot C_\epsilon(y^2)}{C_\epsilon(y)^2}. \end{aligned} \tag{2}$$

Condition **(C1)**, which states that v_2 is $o(1)$, breaks down into three parts (recall that $x := x_n$ and $y := y_n$ are defined respectively by $xB^{'}(x)/B(x) = n = yC^{'}(y)/C(y)$, and vary when $n \to +\infty$):

$$B_\epsilon \odot B_\epsilon(x^2) = o\left(B_\epsilon(x)^2\right); \tag{3}$$

$$B_\epsilon \odot C_\epsilon(xy) = o\left(B_\epsilon(x) C_\epsilon(y)\right); \tag{4}$$

$$C_\epsilon \odot C_\epsilon(y^2) = o\left(C_\epsilon(y)^2\right). \tag{5}$$

We now check that conditions (3) to (5) hold in the different, standard cases that may appear when generating a combinatorial object by Boltzmann sampling. By taking $\mathcal{B} = \mathcal{C}$, it suffices to consider condition (4).

5.5. Evaluation of v_2

Lemma 5.6. Let $\mathcal{B}$ and $\mathcal{C}$ follow a Boltzmann distribution, either flat or bumpy, and let x and y be the approximate values of the Boltzmann parameters ($xB^{'}(x)/B(x) = n = yC^{'}(y)/C(y)$). Then

- If $\mathcal{B}$ and $\mathcal{C}$ are both flat, $B_\epsilon \odot C_\epsilon(xy)/B_\epsilon(x)\, C_\epsilon(y) = \Theta(1/n)$.
- If $\mathcal{B}$ is bumpy and $\mathcal{C}$ is flat, $B_\epsilon \odot C_\epsilon(xy)/B_\epsilon(x)\, C_\epsilon(y) = \Theta(1/n)$.
- If $\mathcal{B}$ and $\mathcal{C}$ are both bumpy, $B_\epsilon \odot C_\epsilon(xy)/B_\epsilon(x)\, C_\epsilon(y) = \Theta(1/\sigma)$ with $\sigma = \min(\sigma_B, \sigma_C)$.

Proof:
Note that, either for a bumpy or flat distribution [4]

$$B_\epsilon(x) = \Theta(B(x)).$$

Of course, a similar relation holds for the class $\mathcal{C}$. Thus the ratio we consider simplifies into $B_\epsilon \odot C_\epsilon(xy)/B(x)\, C(y)$. As we consider either flat or bumpy distributions, we have three cases to consider.

[2] The functions $\mathcal{B}_\epsilon$ and $\mathcal{C}_\epsilon$ actually depend on n through the parameter ϵ; we omit this parameter in our notation for simplicity's sake.

- **If $\mathcal{B}$ and $\mathcal{C}$ both follow a flat distribution.** Intuitively, a flat distribution converges to a uniform distribution when $x \to \rho^-$, and we expect the assumptions of Selivanov's theorem to hold. We now make precise this intuition.

  Equation (1) of Lemma 5.5, applied to both classes, gives

$$\frac{b_m x^m}{B(x)} . \frac{c_m y^m}{C(y)} \sim \frac{\alpha_b^{\alpha_b}\, \alpha_c^{\alpha_c}\, e^{-(\alpha_b+\alpha_c)m/n}}{\Gamma(\alpha_b)\,\Gamma(\alpha_c)} \cdot \left(\frac{m}{n}\right)^{\alpha_b+\alpha_c} \cdot \frac{1}{m^2}.$$

  Hence

$$\begin{aligned} \frac{B_\epsilon \odot C_\epsilon(xy)}{B(x)C(y)} &\sim \sum_{I_\epsilon} \frac{b_m c_m (xy)^m}{B(x)C(y)} \\ &\sim \frac{\alpha_b^{\alpha_b}\, \alpha_c^{\alpha_c}\, e^{-(\alpha_b+\alpha_c)}}{\Gamma(\alpha_b)\,\Gamma(\alpha_c)} \cdot \sum_{m \in I_\epsilon} \left(\frac{m}{n}\right)^{\alpha_b+\alpha_c} \cdot \frac{1}{m^2}. \end{aligned}$$

  As the last sum has exact order $1/n$, the result is proved.

- **If $\mathcal{B}$ follows a bumpy distribution and $\mathcal{C}$ a flat one.** We apply again Lemma 5.5 to obtain

$$\begin{aligned} \frac{B_\epsilon \odot C_\epsilon(xy)}{B(x)C(y)} &= \sum_{m \in I_\epsilon} \frac{b_m x^m}{B(x)} \cdot \frac{c_m y^m}{C(y)} \\ &\sim \frac{\alpha_c^{\alpha_c} e^{-\alpha_c}}{\sqrt{2\pi}\,\Gamma(\alpha_c)\,\sigma_B(x)} \sum_{m \in I_\epsilon} \left(\frac{m}{n}\right)^{\alpha_c} \frac{1}{m} \cdot e^{-(m-n)^2/2\sigma_B^2(x)+o(1)}. \end{aligned}$$

  Now $m = \Theta(n)$ in I_ϵ and we have to approximate $\sum_{m \in I_\epsilon} e^{-(m-n)^2/2\sigma_B^2(x)}$. By taking

$$u = \frac{m-n}{\sigma_B(x)},\ f(u) = e^{-\frac{u^2}{2}} \text{ and } A = \frac{n\epsilon}{\sigma_B},$$

  we have that, with $du = \frac{1}{\sigma_B(x)}$ being the increase in u between two consecutive terms of the sum,

$$\begin{aligned} \sum_{m \in I_\epsilon} e^{-(m-n)^2/2\sigma_B^2(x)} &= \sum_{m \in I_\epsilon} f(u) \\ &= \sigma_B(x) \sum_{m \in I_\epsilon} f(u)\, du \\ &\sim \sigma_B(x) \int_{-A}^{+A} f(u)\, du \\ &\sim \sigma_B(x) \int_{-\infty}^{+\infty} f(u) du = \Theta(\sigma_b), \end{aligned}$$

  the last equivalence coming from $n/\sigma_B \to +\infty$, which comes from the bumpy condition. Putting together all the terms, we obtain that $B_\epsilon \odot C_\epsilon(xy)/B(x)C(y) = \Theta(1/n)$.

- **If $\mathcal{B}$ and $\mathcal{C}$ both follow a bumpy distribution.** Again the approximations provided by Lemma 5.5 give

  $$\frac{B_\epsilon \odot C_\epsilon(xy)}{B(x)C(y)} \quad \sim \quad \frac{1}{\sigma_B\, \sigma_C} \sum_{m \in I_\epsilon} e^{-(m-n)^2(1/\sigma_B^2+1/\sigma_C^2)/2+o(1)}.$$

  Define s by $1/s^2 = 1/\sigma_B^2 + 1/\sigma_C^2$; then

  $$\frac{B_\epsilon \odot C_\epsilon(xy)}{B(x)C(y)} \quad \sim \quad \frac{1}{\sigma_B\, \sigma_C} \sum_{m \in I_\epsilon} e^{-(m-n)^2/2s^2},$$

  and we are back to the integral we computed in the bumpy/flat case, which gives

  $$\sum_{m \in I_\epsilon} e^{-(m-n)^2/2s^2} = \Theta(s)$$

  and $B_\epsilon \odot C_\epsilon(xy)/B(x)C(y) = \Theta(s/\sigma_B\sigma_C)$. The result follows from the fact that, if σ_B and σ_C have the same order, s also has this order; otherwise s has the same order as the smallest of σ_B and σ_C.

□

Lemma 5.7. Let $\mathcal{B}$ and $\mathcal{C}$ follow a Boltzmann distribution, either flat or bumpy, with x and y the approximate values of the Boltzmann parameters, defined by $xB^{'}(x)/B(x) = n = yC^{'}(y)/C(y)$; and let $v_2 = p_1^2 + p_2^2$. Then

- if $\mathcal{B}$ and $\mathcal{C}$ are both flat, $v_2 = \Theta(1/n)$.
- if $\mathcal{B}$ is bumpy and $\mathcal{C}$ is flat, $v_2 = \Theta(1/\sigma_B)$.
- if $\mathcal{B}$ and $\mathcal{C}$ are both bumpy, $v_2 = \Theta(1/\sigma)$ with $\sigma = \min(\sigma_B, \sigma_C)$.

Proof:
Obvious: apply the approximations provided by Lemma 5.6 to the expression of v_2 given by (2) and recall, in the case bumpy/flat, that $\sigma_B = o(n)$. □

5.6. Complexity of the Hadamard sampler

Theorem 5.8. Let $\mathcal{B}$ and $\mathcal{C}$ two combinatorial classes, and define τ_N as the waiting time to draw an object in $\mathcal{B} \odot \mathcal{C}$.

- If both classes $\mathcal{B}$ and $\mathcal{C}$ follow a flat distribution, then the expected time for Algorithm 2 is $\mathbb{E}[\tau_N] = \mathcal{O}(n\sqrt{n})$, and the variance is $\mathcal{O}(n^2)$.
- If $\mathcal{B}$ follows a bumpy distribution and $\mathcal{C}$ a flat one, then the expected time for Algorithm 2 is $\mathbb{E}[\tau_N] = \mathcal{O}\left(n\sigma_B\right)$.
- If both classes $\mathcal{B}$ and $\mathcal{C}$ follow a bumpy distribution, then the expected time for Algorithm 2 is $\mathbb{E}[\tau_N] = \mathcal{O}\left(n \min\left(\sigma_B, \sigma_C\right)\right)$.

Proof:
Lemmas 5.6 and 5.7 show that the conditions of Selivanov's theorem are satisfied with $q_1 = q_2$. Then the normalized waiting time for the first collision $\alpha\tau_N$ follows a Rayleigh distribution with $\alpha = \sqrt{2v_2q_1q_2} = \sqrt{\frac{v_2}{2}}$ and $\mathbb{E}[\tau_n] \sim \sqrt{p_{\max}/4v_2q_1q_2} = \sqrt{p_{\max}/v_2}$. As approximate-size Boltzmann sampling runs in linear expected time, it suffices to compute v_2 in the different cases to obtain the expected number of samples that need to be drawn, thus the expected time of the algorithm. □

5.7. Space complexity

For standard Boltzmann sampling, the space complexity required by the generation algorithm is proportional to the size of the object. This no longer holds for the Hadamard product: the space complexity of our algorithm depends on the expected time for the first collision; under the assumptions of Theorem 3.2 it is of order $\mathcal{O}(n\sqrt{n})$.

Assume that we want to achieve an average size n: we draw objects of $\mathcal{B}$ and $\mathcal{C}$ whose size follows a Boltzmann distribution, and keep those objects whose size belongs to $[(1-\varepsilon)n, (1+\varepsilon)n]$, with ε a fixed parameter. A reasonable value for ε is $0.1 = 10\%$. Of course, we keep only one object of each size for $\mathcal{B}$ and for $\mathcal{C}$. Hence, an upper bound on the number of objects in each class comes from the expected time $E_{\mathcal{A}}$ for the first two-colors collision. We can also obtain a lower bound, by considering same-color collisions: assume that there are respectively $E_{\mathcal{B}}$ and $E_{\mathcal{C}}$ collisions in the classes $\mathcal{B}$ and $\mathcal{C}$ before time $E_{\mathcal{A}}$; then the number of objects in both classes is $2E_{\mathcal{A}} - E_{\mathcal{B}} - E_{\mathcal{C}}$. The precise value of the space needed to store those objects depends on their size, i.e. on the locations of the urns with at least one ball in a coupon-collector problem.

6. Conclusion and extensions

We have presented in this paper a general purpose approximate-size sampler for the Hadamard product. Our sampler works for the classical combinatorial classes, for which it allows us to generate a random Hadamard product in time $O(n\sigma)$, where σ^2 is the smallest of the variances for the two combinatorial classes involved in the product.

We might consider the multivariate Hadamard product, which builds a k-uple of objects sharing the same size. The analysis of the algorithm complexity again requires that of the expected time, which is an extension of the birthday problem to the waiting time until all the k colors appear in a single urn; it requires an extension of Selivanov's results, which deal with the first appearance of two colors in the same urn. Some recent work have been done, generalizing the birthday problem [10]. Preliminary studies seem to indicate a larger expected time, which however remains $o(n^2)$.

The Hadamard product as intermediate constructor. A restriction of the sampler we have presented in this paper is that the Hadamard product appears as the *final* constructor. It would be desirable to extend our sampler so that it allows for the Hadamard product to appear as an intermediate constructor when building complex objects. This requires that the probability distribution for the objects obtained from the Hadamard product follows a Boltzmann distribution, which would require in turn some ”unbiaising” of the random object we obtain.

The rational case. We should mention alternative, possibly more efficient, ways to obtain a Hadamard sampler in special cases. E.g., a classical result of Borel states that the Hadamard product of two rational languages is also a rational language. We consider here how a constructive proof of this result leads to the construction of a sampler for the Hadamard product.

For $i \in \{1, 2\}$, let A_i be a deterministic automaton that recognizes the language L_i. Classically, we denote by E_i its states, $\mathcal{A}_i$ its alphabet, e_i its initial state, T_i its terminal states and Δ_i its transitions. We define the Hadamard product $A_1 \odot A_2$ of A_1 and A_2 as the automaton with states in $E_1 \times E_2$ on the alphabet $\mathcal{A}_1 \times \mathcal{A}_2$, such that there is a transition labelled (a, b) between (e, f) and (e', f') if and only if (e, a, e') and (f, b, f') are both transitions respectively in A_1 and A_2. The initial state is (e_1, e_2) and the set of terminal states is $T_1 \times T_2$. Clearly, the generating function of the language recognized by $A_1 \odot A_2$ is the Hadamard product of the generating functions of the languages L_1 and L_2.

Should we wish to build a Boltzmann generator for the Hadamard product of two rational languages, rather than using our sampler we would build the Hadamard product of the associated deterministic automaton, then to obtain a Boltzmann sampler from its combinatorial specification in the standard way. In particular, using approximate size Boltzmann sampling, the expected time complexity becomes linear in this case. Indeed, a classical approximate size Boltzmann generation on a flat type distribution class is linear (and the Boltzmann distributions associated with rational languages are always flat).

Acknowledgments This work was supported by the French ANR projects Gamma, Boole and Magnum. The authors are grateful to the referees for their perceptive and encouraging comments.

References

[1] Arnold, D., Sleep, M.: Random generation of balanced parenthesis strings, *ACM TOPLAS*, **2**(2), January 1980, 122–128.

[2] Bassino, F., Nicaud, C., Weil, P.: Random generation of finitely generated subgroups of a free group, *International Journal of Algebra and Computation*, **18**(2), 2008, 375–405.

[3] Bodini, O., Jacquot, A.: Boltzmann Samplers for Colored Combinatorial Objects, *Proceedings of GASCOM 2008*, 2008.

[4] Duchon, P., Flajolet, P., Louchard, G., Schaeffer, G.: Boltzmann samplers for the random generation of combinatorial structures, *Combinatorics, Probability and Computing*, **13**(4-5), 2004, 577–625.

[5] Feller, W.: *An introduction to probability theory and its applications*, vol. 1, Wiley & Sons, New York, 1968.

[6] Flajolet, P., Fusy, É., Pivoteau, C.: Boltzmann Sampling of Unlabelled Structures, *Proceedings of the Ninth Workshop on Algorithm Engineering and Experiments and the Fourth Workshop on Analytic Algorithmics and Combinatorics* (D. Appelgate, Ed.), SIAM Press, 2007, Proceedings of the New Orleans Conference.

[7] Flajolet, P., Gardy, D., Thimonier, L.: Birthday Paradox, Coupon Collectors, Caching Algorithms and Self-Organizing Search, *Discrete Applied Mathematics*, **39**, 1992, 207–229.

[8] Flajolet, P., Sedgewick, R.: *Analytic combinatorics*, Cambridge University Press, 2008.

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

[10] Galbraith, S. D., Holmes, M.: A non-uniform birthday problem with applications to discrete logarithms, *IACR Cryptology ePrint Archive*, **2010**, 2010, 616.

[11] Hadamard, J.: Théorème sur les séries entières, *Acta Mathematica*, **22**(1), 1899, 55–63.

[12] Hayman, W.: A generalisation of Stirling's formula, *Journal für die reine und angewandte Mathematik*, **196**, 1956, 67–95.

[13] Nijenhuis, A., Wilf, H.: Combinatorial algorithms, *New York*, 1978.

[14] Nishimura, K., Sibuya, M.: Occupancy with two types of balls, *Annals of the Institute for Statistical Mathematics*, **40**(1), 1988, 77–91.

[15] Popova, T. Y.: Limit theorems in a model of distribution of particles of two types, *SIAM Journal on Theory of Probability and its Applications*, **6**(13), 1968, 511–516.

[16] Rémy, J.: Un procédé itératif de dénombrement d'arbres binaires et son application à leur génération aléatoire, *RAIRO Theoretical Informatics and Applications*, **19**(2), 1985, 179–195.

[17] Selivanov, B.: On the waiting time in a scheme for the random allocation of colored particles, *Diskretnaya Matematika*, **7**(1), 1995, 134–144.