An unbiased pointing operator for unlabeled structures, with applications to counting and sampling

Manuel Bodirsky * Eric Fusy† Mihyun Kang * Stefan Vigerske *

Abstract

We introduce a general method to count and randomly sample unlabeled combinatorial structures. The approach is based on pointing unlabeled structures in an "unbiased" way, i.e., in such a way that a structure of size n gives rise to n pointed structures. We develop a specific Pólya theory for the corresponding pointing operator, and present a sampling framework relying both on the principles of Boltzmann sampling and on Pólya operators. Our method is illustrated on several examples: in each case, we provide enumerative results and efficient random samplers. The approach applies to unlabeled families of plane and non-plane unrooted trees, and tree-like structures in general, but also to cactus graphs, outerplanar graphs, RNA secondary structures, and classes of planar maps.

1 Introduction

Pointing is an important tool to derive decompositions of combinatorial structures, with many applications in enumerative combinatorics. Such decompositions can for instance be used in polynomial-time algorithms that sample structures of a combinatorial class uniformly at random. For the enumeration of *labeled* structures, pointing is reflected by taking the derivative of the corresponding (typically exponential) generating function. In other words, each structure of size n gives rise to n pointed (or rooted) structures. Other important operations on classes of structures are the disjoint union, the product, and the substitution operation – they correspond to addition, multiplication, and composition of the associated generating functions. Together with the usual basic classes of combinatorial structures (finite classes, Set, Sequence, Cycle), this collection of constructions is a powerful device to define all sort of combinatorial families.

If a class of structures can be described by recursive specifications involving pointing, disjoint unions, products, substitutions, and the basic classes, then the techniques of analytic combinatorics can be applied to obtain enumerative results, to study statistical properties of random structures in the class and to derive efficient random samplers. An impressive account for this line of research is [7]. A recent development in the area of random sampling are *Boltzmann samplers* [6], which are an attractive alternative to the recursive method of sampling [8, 16]. Both frameworks provide random generators of *decomposable* combinatorial classes in an automatic way, the generators being uniform (two objects with the same size have same probability) and having low polynomial complexity. The advantage of Boltzmann samplers over the recursive method of sampling, is that they operate in linear time if a small relative tolerance is allowed for the size of the ouput, and they have a small preprocessing cost, making it possible to sample very large structures. Let us mention that a third general sampling framework based on Markov chains covers a wider range of combinatorial classes, as it does not require a decomposition. But the random generators are mostly limited to *approximate* uniformity. Moreover, it is usually difficult to obtain bounds on the rate of convergence to the uniform distribution. See [14] for a general approach dealing with unlabeled structures.

All the results in this paper concern classes of *unlabeled* combinatorial structures, i.e., structures are considered up to isomorphism. Our aim is to provide a general decomposition strategy for unlabeled combinatorial classes. In the case of the class of all graphs, the labeled and the unlabeled model do not differ a lot, which is due to the fact that almost all graphs do not have a non-trivial automorphism (see e.g. [12]). However, for many interesting classes of combinatorial structures, and in fact for all classes studied in this paper, the difference between the labeled and the unlabeled setting matters. To enumerate unlabeled structures, typically the *ordinary* generating function is the appropriate tool. Disjoint unions and products then still correspond to addition and multiplication of the associated generating functions. Correspondingly, Boltzmann samplers for classes described by recursive specifications involving these operations are available [10].

However, for unlabeled structures the substitution operation no longer corresponds to the composition of

*Humboldt-Universität zu Berlin, Institute für Informatik und Mathematik, Unter den Linden 6, 10099 Berlin, Germany

†Projet Algo, INRIA Rocquencourt, B. P. 105, 78153 Le Chesnay Cedex, France

generating functions, due to the symmetries an unlabeled structure might have. This problem is solved by Pólya theory, where *cycle index sums* are introduced, which extend generating functions and take care of potential symmetries. Pólya theory provides a computation rule for the cycle index sum associated to a substitution construction. The presence of symmetries also entails problems with the *pointing* operator: the fundamental property that a structure of size n gives rise to n pointed (or rooted) structures does not hold. Indeed, if a structure of size n has a non-trivial automorphism, then it gives rise to less than n pointed structures (because rooting at two vertices in symmetric position produces the same rooted structure). Thus, for unlabeled structures, the classical pointing (or rooting) operator does not correspond to the derivative of the ordinary generating function.

In this paper, we introduce an *unbiased* pointing operator for unlabeled structures. It is unbiased in the sense that a structure of size n gives rise to n pointed structures. The operator consists in pointing not only an atom but a *cycle* of atoms associated with a symmetry of the structure. Precisely, a *cycle-pointed* structure is a combinatorial structure A together with a marked cyclic sequence of atoms of A that is a cycle of an automorphism of A. Accordingly, we call our operator *cycle-pointing*. The idea is inspired by Pólya theory and is related to a result of group theory known as Parker's lemma [5]. We develop techniques to apply this new pointing operator to enumeration and random sampling of unlabeled combinatorial classes. The crucial point is that cycle-pointing is *unbiased*. As a consequence, performing both tasks of enumeration and uniform random sampling on a combinatorial class is equivalent to performing these tasks on the associated cycle-pointed class.

To understand the way in which we use our operator, it is instructive to look at the class of *free trees*, i.e., unrooted and nonplane *trees* (equivalently, acyclic connected graphs). Building on the work of Cayley and Pólya, Otter [17] determined the exact and asymptotic number of free trees. To this end, he developed the by-now standard dissimilarity characteristic equation, which relates the number of free trees with the number of rooted nonplane trees; see [11]. The best-known method to sample free trees uniformly at random is due to Wilf [20], and uses the concept of the centroid of a tree. The method is an example of application of the recursive method of sampling and requires a preprocessing step to compute a table of quadratic size.

Cycle-pointing provides a new way to count and sample free trees. Both tasks are carried out on cycle-pointed (nonplane) trees. The advantage of studying cycle-pointed trees is that the pointed cycle gives a starting point for a recursive decomposition. In the case of cycle-pointed trees, we can formulate such a decomposition using the standard operations of disjoint union, product, and substitution (these operations being defined on cycle-pointed structures). We want to stress that, despite some superficial similarities, this method of enumerating free trees is fundamentally different from the previously existing methods mentioned above. Indeed, the dissimilarity characteristic equation leads to generating function equations involving *subtraction*. The existence of subtraction yields massive rejection when translated to a random generator for the class of structures (e.g., using Boltzmann samplers or the recursive method). In contrast, the equations produced by the method based on cycle-pointing have only positive signs, and the existence of a Boltzmann sampler for free trees, with no rejection involved, will follow directly from the general results derived in this paper. As usual, the Boltzmann samplers we obtain have a running time that is linear in the size of the structure generated, and have small pre-processing cost.

Similarly, we can decompose plane and non-plane trees, and more generally all sorts of tree-like structures, such as RNA secondary structures. By the observation that the block decomposition of a graph has also a tree-like structure, we can apply the method to classes of graphs where the two-connected components can be explicitly enumerated. This leads to efficient Boltzmann samplers for instance for cactus graphs and outerplanar graphs, improving on the generators of [4]. Finally, we sketch how the method can be applied to count and sample certain classes of planar maps. This demonstrates that our strategy is not limited to tree-like structures, but can also be applied to other classes of structures that allow for a recursive decomposition.

2 Preliminaries

This paper deals with the enumeration and random generation of families of *combinatorial structures* [11], such as graphs, trees, plane trees, maps, etc. These structures are constituted of *atoms* (e.g. nodes of a tree, vertices of a graph). The size of a structure is defined as its number of atoms. It is often convenient to use the same letter A for a combinatorial structure and for its set of atoms; for example, we write $v \in G$ if v is a vertex of a graph G. Two structures A and B are *isomorphic* if there exists a bijection σ from A to B that is structure-preserving; σ is then called an *isomorphism* between A and B. An isomorphism between A and itself is called an *automophism*, and the set of automorphisms of A is denoted by $\mathrm{Aut}(A)$. From now on, all combinatorial structures are assumed to be *well labeled*, i.e., a structure of size n has $[1..n]$ as its set of atoms.

We are interested in *unlabeled* structures, i.e., we consider the structures up to isomorphism. For enumeration, this means that we count *isomorphism classes* of a family. These isomorphism classes can be viewed as *unlabeled* combinatorial structures. For random generation, we are looking for efficient algorithms that produce, given a number n, a structure from this family such that each isomorphism class of structures (i.e., each unlabeled structure) with n atoms is selected with the same probability.

If $\mathcal{A}$ is a family of combinatorial structures (also called a combinatorial class), $\mathcal{A}_n$ denotes the subset of structures of $\mathcal{A}$ with exactly n atoms, and $a_n := |\mathcal{A}_n|$ denotes the number of such structures. Correspondingly, we use $\widetilde{\mathcal{A}}$, $\widetilde{\mathcal{A}}_n$, and $\tilde{a}_n$ for unlabeled combinatorial structures. The series $a(x) := \sum_{n\geq 0} \frac{1}{n!} a_n x^n$ is called the exponential generating function of $\mathcal{A}$, and the series $\tilde{a}(x) := \sum_{n\geq 0} \tilde{a}_n x^n$ is called the ordinary generating function of $\widetilde{\mathcal{A}}$.

2.1 Cycle index sums. For each $n \geq 0$, a *symmetry* of $\mathcal{A}$ of size n is a pair (A, σ) where A is from $\mathcal{A}_n$ and σ is an automorphism of A. Notice that the automorphism σ can be the identity. The set of all symmetries (A, σ) where $A \in \mathcal{A}$ is denoted by $\mathcal{S}(\mathcal{A})$. The *weight-monomial* of a symmetry (A, σ) of size n is defined as

$$w_{(A,\sigma)} := \frac{1}{n!} \prod_{i=1}^{n} s_i^{c_i}, \tag{2.1}$$

where c_i is the number of cycles of σ of length i, and the s_i are formal variables. The *cycle index sum* of $\mathcal{A}$, denoted by $Z_{\mathcal{A}}(s_1, s_2, \ldots)$, or shortly $Z_{\mathcal{A}}$, is the formal power series defined as the sum of the weight-monomials of the symmetries of $\mathcal{A}$:

$$Z_{\mathcal{A}}(s_1, s_2, \ldots) := \sum_{(A,\sigma)\in\mathcal{S}(\mathcal{A})} w_{(A,\sigma)}. \tag{2.2}$$

Cycle index sums have been introduced by Pólya. A detailed definition and many examples and applications can be found in [2] and [11]. The following lemma, which relies on Burnside's lemma, ensures that cycle index sums are a refinement of ordinary generating functions.

Lemma 2.1. (Pólya) *Let $\mathcal{A}$ be a combinatorial class. For $n \geq 0$, each unlabeled structure $\tilde{A} \in \widetilde{\mathcal{A}}_n$ gives rise to $n!$ symmetries, i.e., there are $n!$ symmetries (A, σ) such that the unlabeled structure of A is $\tilde{A}$. As a consequence,*

$$Z_{\mathcal{A}}(x, x^2, x^3, \ldots) = \tilde{a}(x). \tag{2.3}$$

2.2 Combinatorial Constructions. We now recall fundamental constructions to define combinatorial classes from other classes. We start with the description of some basic classes, and then introduce the three constructions: disjoint union, product, and substitution.

Basic classes. Basic finite classes are the class ε consisting of one structure of size 0, with cycle index sum $Z_\varepsilon = 1$, and the *atom class* $\mathcal{Z}$ consisting of one structure of size 1, with cycle index sum $Z_{\mathcal{Z}} = s_1$. The class Seq consists of structures that are sequences of atoms (i.e., the atoms are linearly ordered). It has cycle index sum $Z_{\text{Seq}} = 1/(1 - s_1)$; similarly, the class Cyc consists of structures that are non-empty cyclic sequences of atoms, having cycle index sum $Z_{\text{Cyc}} = \sum_{i\geq 1} \phi(i)/i \cdot \log(1/(1-s_i))$; finally, the class Set consists of all structures that are sets of atoms. It has cycle index sum $Z_{\text{Set}} = \exp(\sum_{i\geq 1} s_i/i)$ (the expressions of cycle index sums are given in [2]).

Disjoint union. The *disjoint union* (also called sum) of two classes $\mathcal{A}$ and $\mathcal{B}$ is denoted by $\mathcal{A} + \mathcal{B}$. If $\mathcal{A}$ and $\mathcal{B}$ are setwise not disjoint, we implicitly make them disjoint by attaching one label to all members of $\mathcal{A}$ and a different label to all members of $\mathcal{B}$.

Product. The cartesian (often also called *partitional* or *dinary*) product $\mathcal{M} = \mathcal{A} \times \mathcal{B}$ of two classes $\mathcal{A}$ and $\mathcal{B}$ is the set of all pairs $M = (A, B)$, where A is isomorphic to a structure of $\mathcal{A}$ and B is isomorphic to a structure of $\mathcal{B}$ (and the atoms of M are $[1, \ldots, |A| + |B|]$).

Substitution. Given two classes $\mathcal{A}$ and $\mathcal{B}$ such that $\mathcal{B}$ has no structure of size 0, $\mathcal{M} = \mathcal{A} \circ \mathcal{B}$ is the family of combinatorial structures M that can be obtained by replacing each atom v of a structure of $\mathcal{A}$ (called the *core structure*) by a structure B_v isomorphic to a structure of $\mathcal{B}$. Again M is well labeled, so that the atoms of M are $[1.. \sum_{v\in A} |B_v|]$.

Together with the basic classes, these constructions provide an extremely powerful device for the description of combinatorial families. We can also express other combinatorial constructions, such as the formation of sequences, cycles, and sets of structures of $\mathcal{A}$, which are specified as Seq $\circ\, \mathcal{A}$, Cyc $\circ\, \mathcal{A}$, and Set $\circ\, \mathcal{A}$, respectively.

Proposition 2.1. (Pólya, Bergeron et al.) *For each of the constructions Sum, Product, and Substitution, there is an explicit rule to compute the associated cycle index sum:*

$$Z_{\mathcal{A}+\mathcal{B}} = Z_{\mathcal{A}} + Z_{\mathcal{B}} \tag{2.4}$$

$$Z_{\mathcal{A}\times\mathcal{B}} = Z_{\mathcal{A}} \cdot Z_{\mathcal{B}} \tag{2.5}$$

$$Z_{\mathcal{A}\circ\mathcal{B}} = Z_{\mathcal{A}}(Z_{\mathcal{B}}(s_1, s_2, \ldots), Z_{\mathcal{B}}(s_2, s_4, \ldots), \ldots). \tag{2.6}$$

3 Cycle-pointed Classes

In this section we introduce the concept of *cycle-pointed families*, which allows us to define our unbiased pointing operator. A *cycle-pointed* structure $P = (A, C)$ is a combinatorial structure A together with a distinguished cycle of atoms $C = (v_1, \ldots, v_l)$ such that there exists at least one automorphism of A having C as one of its cycles (i.e., $(v_1, \ldots, v_l)$ is mapped to $(v_2, \ldots, v_l, v_1)$). The cycle C is called the *marked cycle* (or pointed cycle) of P, and A is called the *underlying structure* of P. An automorphism σ of A having C as one of its cycles is called a *C-automorphism* of P, and the other cycles of σ are called *unmarked*.

A *cycle-pointed combinatorial family* $\mathcal{P}$ is a class of cycle-pointed structures. As for classical structures, the size of a structure $P \in \mathcal{P}$ is its number of atoms. For $l \geq 1$, we denote by $\mathcal{P}_{(l)}$ the set of structures of $\mathcal{P}$ whose marked cycle has length l. For $n \geq 0$, the set of structures of $\mathcal{P}$ of size n is denoted by $\mathcal{P}_n$. Two cycle-pointed structures P and P' of $\mathcal{P}$ are *isomorphic* if there exists an isomorphism from the underlying unpointed structure of P to the underlying unpointed structure of P' that maps the marked cycle of P to the marked cycle of P' (i.e., the marked cycle of P is mapped to the marked cycle of P' in such a way that the cyclic order is preserved). We denote by $\widetilde{\mathcal{P}}$ the class of structures of $\mathcal{P}$ considered up to isomorphism. The class $\widetilde{\mathcal{P}}$ is called the *unlabeled class* of $\mathcal{P}$ and we define the ordinary generating function of $\widetilde{\mathcal{P}}$ as $\tilde{p}(x) := \sum_n |\widetilde{\mathcal{P}}_n| x^n$.

To develop Pólya theory for cycle-pointed classes, we introduce the terminology of *C-symmetry* and *rooted C-symmetry*. Given a cycle-pointed class $\mathcal{P}$, a *C-symmetry* on $\mathcal{P}$ is a pair (P, σ) where $P \in \mathcal{P}$ and σ is a C-automorphism of P. A *rooted C-symmetry* is a triple (P, σ, v), where (P, σ) is a C-symmetry, and v is one of the atoms of the marked cycle of P; this atom is called the *root* of the rooted C-symmetry. The set of rooted C-symmetries of $\mathcal{P}$ is denoted by $\mathcal{R}(\mathcal{P})$.

The *weight-monomial* of a rooted C-symmetry of size n is defined as

$$w_{(P,\sigma,v)} := \frac{1}{n!} t_l \prod_{i=1}^{n} s_i^{c_i}, \tag{3.7}$$

where l is the length of the marked cycle, t_l and $s_1, \ldots, s_n$ are formal variables, and c_i is the number of unmarked cycles of σ of length i. We define the *cycle index sum* $Z_{\mathcal{P}}(s_1, t_1; s_2, t_2; \ldots)$ of $\mathcal{P}$ (shortly written $Z_{\mathcal{P}}$) as the sum of the weight-monomials over all rooted C-symmetries of $\mathcal{P}$:

$$Z_{\mathcal{P}}(s_1, t_1; s_2, t_2; \ldots) := \sum_{(P,\sigma,v) \in \mathcal{R}(\mathcal{P})} w_{(P,\sigma,v)}. \tag{3.8}$$

The following lemma is the counterpart of Lemma 2.1 for cycle-pointed classes.

Lemma 3.1. *Let $\mathcal{P}$ be a cycle-pointed class. For $n \geq 0$, each unlabeled structure $\tilde{P} \in \widetilde{\mathcal{P}}$ of size n gives rise to exactly $n!$ rooted C-symmetries, i.e., there are $n!$ rooted C-symmetries (P, σ, v) such that the unlabeled structure of P is $\tilde{P}$. As a consequence,*

$$\tilde{p}(x) = Z_{\mathcal{P}}(x, x; x^2, x^2; \ldots). \tag{3.9}$$

3.1 Combinatorial Constructions.

We now adapt the constructions of disjoint union, product and substitution to cycle-pointed structures.

Disjoint union. The definition is the same as in the unpointed case. The *disjoint union* (or sum) of two cycle-pointed classes $\mathcal{P}$ and $\mathcal{Q}$, denoted by $\mathcal{P} + \mathcal{Q}$, is the union of $\mathcal{P}$ and $\mathcal{Q}$, where we again assume that $\mathcal{P}$ and $\mathcal{Q}$ are disjoint sets.

Product. Given a cycle-pointed class $\mathcal{P}$ and a non cycle-pointed class $\mathcal{A}$, the family $\mathcal{P} \times \mathcal{A}$ ($\mathcal{A} \times \mathcal{P}$) consists of all pairs (P, A) ((A, P) respectively) where P is isomorphic to a structure of $\mathcal{P}$ and A is isomorphic to a structure of $\mathcal{A}$. Note that exactly one of the two families has to be cycle-pointed to produce a cycle-pointed family (i.e., the product of two cycle-pointed classes is not allowed).

Cycle-pointed substitution. To describe cycle-pointed substitution, we need to define *compositions of cycles*. If $C = (v_1, \ldots, v_l)$ is a cycle of atoms, and $C_1, \ldots, C_k$ is a sequence of k copies of C, then the *composed cycle* of $C_1, \ldots, C_k$ is the cycle of atoms of length lk such that, for $1 \leq i < k$ and $1 \leq j \leq l$, the follower of the atom v_j in C_i is the atom v_j in C_{i+1}; and for $1 \leq j \leq l$, the follower of the atom v_j in C_k is the atom $v_{(j+1) \bmod l}$ in C_1.

Let $\mathcal{P}$ and $\mathcal{Q}$ be cycle-pointed classes and $\mathcal{A}$ a non cycle-pointed class, such that $\mathcal{A}$ and $\mathcal{Q}$ have no object of size 0. Then $\mathcal{P} \odot (\mathcal{A}, \mathcal{Q})$ is the family of cycle-pointed structures that can be obtained as follows (see Figure 1 for an example).

1. Take a structure $P \in \mathcal{P}$ and let $C = (u_1, \ldots, u_k)$ be the marked cycle of P. (We implicitly choose a *starting vertex* u_1 in the cycle.)

2. Substitute the atoms of C by structures of $\mathcal{Q}$ and the unmarked atoms of P by structures of $\mathcal{A}$, so as to respect a symmetry induced by C. In other words, there exists a C-automorphism Γ of P such that, for each atom v of P, the structures substituted at v and at $\Gamma(v)$ are isomorphic. In particular, the structures substituted at $u_1, \ldots, u_k$ are isomorphic copies of a structure $Q \in \mathcal{Q}$.

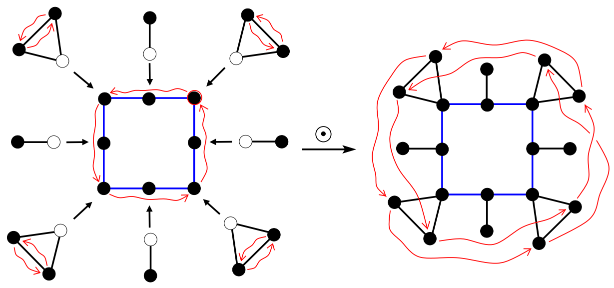

Figure 1: A cycle-pointed structure obtained from a substitution.

3. Let $C_1, \ldots, C_k$ be the marked cycles of the copies of Q substituted respectively at $u_1, \ldots, u_k$. Then the marked cycle of the composed structure is the composed cycle D of $C_1, \ldots, C_k$.

It is easily shown that there exists a symmetry of the composed structure having D as one of its cycles, as required by the definition of cycle-pointed structures.

We now define a composition of cycle index sums that reflects cycle-pointed substitution. The *cycle-pointed plesthystic composition* $Z_{\mathcal{P}} \odot (Z_{\mathcal{A}}, Z_{\mathcal{Q}})$ is the formal power series

$$Z_{\mathcal{P}} \odot (Z_{\mathcal{A}}, Z_{\mathcal{Q}}) := Z_{\mathcal{P}}(s_i \to Z_{\mathcal{A}}^{[i]}, t_l \to Z_{\mathcal{Q}}^{[l]}),$$

with $Z_{\mathcal{A}}^{[i]} := Z_{\mathcal{A}}(s_i, s_{2i}, \ldots)$ and $Z_{\mathcal{Q}}^{[l]} := Z_{\mathcal{Q}}(s_l, t_l; s_{2l}, t_{2l}; \ldots)$. In other words, each monomial $t_l s_1^{c_1} s_2^{c_2} \ldots s_r^{c_r}$ of $Z_{\mathcal{P}}$ is replaced by $Z_{\mathcal{Q}}^{[l]} (Z_{\mathcal{A}})^{c_1} \left(Z_{\mathcal{A}}^{[2]}\right)^{c_2} \ldots \left(Z_{\mathcal{A}}^{[r]}\right)^{c_r}$.

Proposition 3.1. (Computation rules) *For each of the constructions Sum, Product, and cycle-pointed Substitution, there is an explicit rule to compute the associated cycle index sum,*

(3.10)	$Z_{\mathcal{P}+\mathcal{Q}} = Z_{\mathcal{P}} + Z_{\mathcal{Q}}$	[Sum]
(3.11)	$Z_{\mathcal{P}\times\mathcal{A}} = Z_{\mathcal{A}\times\mathcal{P}} = Z_{\mathcal{A}} \cdot Z_{\mathcal{P}}$	[Product]
(3.12)	$Z_{\mathcal{P}\odot(\mathcal{A},\mathcal{Q})} = Z_{\mathcal{P}} \odot (Z_{\mathcal{A}}, Z_{\mathcal{Q}})$.	[Substitution]

3.2 Pointing. Given a combinatorial class $\mathcal{A}$, we define its *cycle-pointed class* $\mathcal{A}^\bullet$ to be the set of all possible cycle-pointed structures P where the underlying unpointed structure is in $\mathcal{A}$.

Observe that a rooted C-symmetry of $\mathcal{A}^\bullet$ is obtained from a symmetry (A, σ) of $\mathcal{A}$ by choosing an atom v of A and marking the cycle of σ containing v. Equivalently, it is obtained by marking a cycle of atoms corresponding to a cycle of σ and choosing an atom of the cycle as the root of the rooted C-symmetry. The first observation shows that each symmetry (A, σ) of $\mathcal{A}_n$ gives rise to n rooted C-symmetries of $\mathcal{A}_n^\bullet$. The second observation yields

$$(3.13) \qquad Z_{\mathcal{A}^\bullet_{(l)}} = l t_l \frac{\partial}{\partial s_l} Z_{\mathcal{A}}, \quad \text{for each } l \geq 1.$$

For the particular case $l = 1$, corresponding to structures of $\mathcal{A}$ with a unique distinguished vertex (the *root*), we recover the well-known equation relating the cycle index sum of a combinatorial class and of the associated *rooted* class; see [2, Sec.1.4.] and [11].

Theorem 3.1. (unbiased pointing) *Let $\mathcal{A}$ be a combinatorial class. Then, for $n \geq 0$, each unlabeled structure of $\widetilde{\mathcal{A}}_n$ gives rise to exactly n non-isomorphic structures in $\widetilde{\mathcal{A}}_n^\bullet$. Hence, the ordinary generating functions $\tilde{a}(x)$ of $\widetilde{\mathcal{A}}$ and $\tilde{a}^\bullet(x)$ of $\widetilde{\mathcal{A}}^\bullet$ satisfy the relation*

$$(3.14) \qquad \tilde{a}^\bullet(x) = x \frac{\mathrm{d}}{\mathrm{d}x} \tilde{a}(x).$$

Proof. Given $\tilde{A} \in \widetilde{\mathcal{A}}_n$, let S be the set of structures of $\widetilde{\mathcal{A}}^\bullet$ whose underlying structure (obtained by unmarking the pointed cycle) is $\tilde{A}$. Thus the proof of the lemma reduces to proving that S has cardinality n. Let $\mathcal{S}(\tilde{A})$ be the set of symmetries for the structure $\tilde{A}$, and let $\mathcal{R}(S)$ be the set of rooted C-symmetries for structures from S. Lemma 2.1 ensures that $\tilde{A}$ gives rise to $n!$ symmetries and Lemma 3.1 ensures that each structure of $\widetilde{\mathcal{A}}_n^\bullet$ gives rise to $n!$ rooted C-symmetries. Hence $|\mathcal{S}(\tilde{A})| = n!$ and $|\mathcal{R}(S)| = |S|n!$. In addition, we have seen that a symmetry (A, σ) of $\mathcal{A}_n$ gives rise to n rooted C-symmetries of $\mathcal{A}_n^\bullet$. Hence, $|\mathcal{R}(S)| = n|\mathcal{S}(\tilde{A})|$. Since $|\mathcal{S}(\tilde{A})| = n!$ and $|\mathcal{R}(S)| = |S|n!$, we obtain $|S|n! = nn!$, and therefore $|S| = n$.

Our pointing operator is easily injected in the three operations Sum, Product, and Substitution. By construction, we have the three following rules:

(3.15)	$(\mathcal{A} + \mathcal{B})^\bullet$	$=$	$\mathcal{A}^\bullet + \mathcal{B}^\bullet$
(3.16)	$(\mathcal{A} \times \mathcal{B})^\bullet$	$=$	$\mathcal{A}^\bullet \times \mathcal{B} \; + \; \mathcal{A} \times \mathcal{B}^\bullet$
(3.17)	$(\mathcal{A} \circ \mathcal{B})^\bullet$	$=$	$\mathcal{A}^\bullet \odot (\mathcal{B}, \mathcal{B}^\bullet)$.

4 Applications

A cycle-pointed family is partitioned into structures with pointed cycle of size 1 (corresponding to *rooted* structures, i.e., with a unique pointed atom), and structures with a pointed cycle of length at least 2. These are called *symmetric cycle-pointed structures*, because the cycle is associated with a nontrivial automorphism. The subclass of symmetric cycle-pointed structures in a cycle-pointed class $\mathcal{P}$ is denoted by $\mathcal{P}_{\geq 2}$. In all examples given here, the strategy of decomposition of cycle-pointed structures is the following: the decomposition of rooted structures starts at the root; and the decomposition of symmetric cycle-pointed structures starts at a so-called *centre of symmetry*.

4.1 Trees. We first illustrate the method on unrooted trees, starting with the precise definition of centre of symmetry. Let T be a symmetric cycle-pointed tree, and let $(v_1, \ldots, v_l)$ be the marked cycle of T ($l \geq 2$). For $1 \leq i \leq l$, let P_i be the path of T joining v_i to $v_{(i+1) \bmod l}$. Then it is easily shown that the paths P_i share the same middle. If the paths P_i have odd (even) length, then the common middle is the middle of an edge e (is a vertex v, respectively). In the first (second) case e (v, respectively) is called the *centre of symmetry* of T. Clearly, the centre of symmetry is fixed by any C-automorphism of T.

Let $\mathcal{F}$ be the class of *free trees*, i.e., unrooted nonplane trees (equivalently, acyclic connected graphs), and let $\mathcal{R}$ be the class of rooted nonplane trees. Rooted nonplane trees are decomposed at the root. As the children of the root node are unordered, we have $\mathcal{R} = \mathcal{Z} \times \textsc{Set} \circ \mathcal{R}$. Symmetric cycle-pointed trees are decomposed at the centre of symmetry, which can be an edge or a vertex. In the first case, the structure can be described as a cycle-pointed substitution in the one-element cycle-pointed class $\mathcal{L}_2^\bullet$, where $\mathcal{L}_2^\bullet$ consists of the edge-graph e endowed with a pointed cycle exchanging the two extremities of e. In the second case, let the vertex v be the centre of symmetry. Then a collection of at least 2 isomorphic copies of a cycle-pointed rooted tree are attached to v at their root, and other rooted nonplane trees (with no pointed cycle) may also be attached at v. This yields $\mathcal{F}^\bullet_{\geq 2} = \mathcal{L}_2^\bullet \odot (\mathcal{R}, \mathcal{R}^\bullet) + \mathcal{Z} \times \textsc{Set}^\bullet_{\geq 2} \odot (\mathcal{R}, \mathcal{R}^\bullet)$. To summarize, the class $\mathcal{F}^\bullet$ has the following 4-lines decomposition grammar,

(4.18)

$$\left\{ \begin{array}{rcl} \mathcal{F}^\bullet &=& \mathcal{R} + \mathcal{F}^\bullet_{\geq 2} \\ \mathcal{R} &=& \mathcal{Z} \times \textsc{Set} \circ \mathcal{R} \\ \mathcal{F}^\bullet_{\geq 2} &=& \mathcal{L}_2^\bullet \odot (\mathcal{R}, \mathcal{R}^\bullet) + \mathcal{Z} \times \textsc{Set}^\bullet_{\geq 2} \odot (\mathcal{R}, \mathcal{R}^\bullet) \\ \mathcal{R}^\bullet &=& \mathcal{Z}^\bullet \times \textsc{Set} \circ \mathcal{R} + \mathcal{Z} \times \textsc{Set}^\bullet \odot (\mathcal{R}, \mathcal{R}^\bullet), \end{array} \right.$$

where the 4th line, which is necessary to make the grammar completely recursive, is obtained from the second line using the derivative rules (3.15), (3.16), and (3.17). Let $f(x)$ and $R(x)$ be the ordinary generating functions of unlabeled free trees and unlabeled rooted nonplane trees. Then, the decomposition grammar (4.18) translates —via the computation rules of Pólya operators and then the specialization $(s_i \to x^i, t_i \to x^i)$— to

(4.19)

$$\left\{ \begin{array}{rcl} R(x) &=& x \exp\left(\sum_{i \geq 1} \frac{1}{i} R(x^i)\right) \\ x f'(x) &=& R(x) + x^2 R'(x^2) + \left(\sum_{l \geq 2} x^l R'(x^l)\right) R(x). \end{array} \right.$$

Using $xR'(x) = R(x)(1 + \sum_{l \geq 1} x^l R'(x^l))$, this simplifies to $xf'(x) = xR'(x)(1 - R(x)) + x^2 R'(x^2)$, which agrees with Otter's formula $f(x) = R(x) - \frac{1}{2}(R^2(x) - R(x^2))$. The main difference between our method and Otter's dissimilarity method is that our expressions (4.19) have only positive signs, as they reflect a decomposition grammar. This is crucial to obtain random generators without rejection.

Our method works as well with unrooted *plane* trees, yielding a similar 4-lines decomposition grammar. The only difference is that the neighbours of a node are cyclically ordered, so that the $\textsc{Set}$ construction is replaced by a $\textsc{Seq}$ construction for rooted trees, and the $\textsc{Set}^\bullet_{\geq 2}$ is replaced by a $\textsc{Cyc}^\bullet_{\geq 2}$ construction for symmetric cycle-pointed trees. The same technique can also be applied to enumerate any family of unrooted nonplane or plane trees where the degrees of the nodes are constrained to be in a finite integer set Ω containing 0. In particular unrooted plane and nonplane binary trees and ternary trees can be enumerated. Notice that the case of unrooted nonplane ternary trees has attracted a lot of attention, as it corresponds to the number of acyclic carbon alkanes [18]. Finally, RNA secondary structures have been given a tree-like decomposition grammar [13], so that our techniques also easily apply in this case.

4.2 Graphs. In the more general case of a family $\mathcal{C}$ of connected graphs, there is a decomposition, called *block decomposition* (see e.g. [11, p.10]), of the graphs of $\mathcal{C}$ into 2-connected components that are articulated at separating vertices of the connected graph. (A separating vertex is a vertex whose removal disconnects the graph; a graph is called 2-connected if it has no separating vertex.) This decomposition yields a bicolored tree with nodes corresponding to 2-connected components and nodes corresponding to vertices, where separating vertices have degree at least 2. We denote by $\widehat{\mathcal{C}}$ the family of graphs of $\mathcal{C}$ rooted at a vertex that does not count in the size, and we denote by $\mathcal{D}$ ($\widehat{\mathcal{D}}$) the family of graphs of $\mathcal{C}$ ($\widehat{\mathcal{C}}$, respectively) that are 2-connected. A structure of $\widehat{\mathcal{C}}$ is called *simple* if it has at least two vertices and if the root vertex is not separating. The family of simple structures of $\widehat{\mathcal{C}}$ is denoted by $\widehat{\mathcal{C}}_s$. Then, decomposing structures of $\widehat{\mathcal{C}}$ at the root yields $\widehat{\mathcal{C}} = \textsc{Set} \circ \widehat{\mathcal{C}}_s$, $\widehat{\mathcal{C}}_s = \widehat{\mathcal{D}} \circ (\mathcal{Z} \times \widehat{\mathcal{C}})$.

Given a symmetric cycle-pointed graph in $\mathcal{C}^\bullet_{\geq 2}$, we consider the induced symmetric cycle-pointed bicolored tree T given by the block-decomposition. As T is bicolored, the centre of symmetry of T can not be an edge (because a C-automorphism of T would exchange the two extremities of the edge, which have different colors). Hence, the centre of symmetry of T is a node

of T, i.e., it is either a 2-connected component or a separating vertex of the connected graph. Accordingly, it is called the centre of symmetry of the connected graph. Decomposing structures of $\mathcal{C}^{\bullet}_{\geq 2}$ at the centre of symmetry yields
$\mathcal{C}^{\bullet}_{\geq 2} = \mathcal{D}^{\bullet}_{\geq 2} \odot (\mathcal{Z} \times \widehat{\mathcal{C}}, (\mathcal{Z} \times \widehat{\mathcal{C}})^{\bullet}) + \mathcal{Z} \times \textsc{Set}^{\bullet}_{\geq 2} \odot (\widehat{\mathcal{C}}_s, \widehat{\mathcal{C}}^{\bullet}_s)$.
Finally, we obtain the following decomposition grammar for $\mathcal{C}^{\bullet}$ in terms of $\mathcal{D}$ and the classes derivated from $\mathcal{D}$:

(4.20)

$$\begin{cases} \mathcal{C}^{\bullet} &= \mathcal{Z}^{\bullet} \times \widehat{\mathcal{C}} + \mathcal{C}^{\bullet}_{\geq 2} \\ \widehat{\mathcal{C}} &= \textsc{Set} \circ \widehat{\mathcal{C}}_s, \quad \widehat{\mathcal{C}}_s = \widehat{\mathcal{D}} \circ (\mathcal{Z} \times \widehat{\mathcal{C}}) \\ \mathcal{C}^{\bullet}_{\geq 2} &= \mathcal{D}^{\bullet}_{\geq 2} \odot (\mathcal{Z} \times \widehat{\mathcal{C}}, (\mathcal{Z} \times \widehat{\mathcal{C}})^{\bullet}) + \mathcal{Z} \times \textsc{Set}^{\bullet}_{\geq 2} \odot (\widehat{\mathcal{C}}_s, \widehat{\mathcal{C}}^{\bullet}_s) \\ \widehat{\mathcal{C}}^{\bullet} &= \textsc{Set}^{\bullet} \odot (\widehat{\mathcal{C}}_s, \widehat{\mathcal{C}}^{\bullet}_s), \quad \widehat{\mathcal{C}}^{\bullet}_s = \widehat{\mathcal{D}}^{\bullet} \odot (\mathcal{Z} \times \widehat{\mathcal{C}}, (\mathcal{Z} \times \widehat{\mathcal{C}})^{\bullet}). \end{cases}$$

If the class $\mathcal{D}$ and its rooted and cycle-pointed classes are combinatorially tractable (i.e., the cycle index sums have explicit expressions), then the decomposition grammar (4.20) translates —via the computation rules of Pólya operators and the specialization $(s_i \to x^i, t_i \to x^i)$— to an equation system on ordinary generating functions similar to (4.19), from which the coefficients counting unlabeled structures of $\mathcal{C}$ according to the size can be extracted. Many classes of graphs have simple 2-connected components: e.g. free trees are such that $\mathcal{D}$ is the edge-graph, and cactus graphs are such that $\mathcal{D} = \mathrm{Cyc}$. For outerplanar graphs (graphs embeddable in the plane so that all vertices are incident to the outer face), the class $\mathcal{D}$ consists of dissections of convex polygons and is combinatorially tractable [3].

4.3 Maps. A map is a planar graph embedded on a sphere up to isotopic deformation, i.e., it is a planar graph together with a cyclic order of the neighbours around each vertex. Maps have motivated a huge literature since the pioneer works of Tutte [19]. Due to the rigidity given by the embedding, the decomposition grammar (4.20) simplifies: there is one equation for each length $l \geq 1$ of the pointed cycle. Taking half-edges as atoms (instead of vertices), we obtain

(4.21)

$$\begin{cases} \widehat{\mathcal{C}} &= \textsc{Seq} \circ \widehat{\mathcal{C}}_s, \quad \widehat{\mathcal{C}}_s = \widehat{\mathcal{D}} \circ (\mathcal{Z} \times \widehat{\mathcal{C}}) \\ \mathcal{C}^{\bullet}_{(l)} &= \mathcal{D}^{\bullet}_{(l)} \odot (\widehat{\mathcal{C}}, \widehat{\mathcal{C}}^{\bullet}) + \textsc{Cyc}^{\bullet}_{(l)} \odot (\widehat{\mathcal{C}}_s, \widehat{\mathcal{C}}^{\bullet}_s), \text{ for } l \geq 2. \end{cases}$$

In the case of maps, the decomposition grammar is used in the other direction. Indeed, unconstrained maps are easy to count using the quotient method [15]. Then, Grammar (4.21) translates to equations from which the coefficients counting 2-connected maps can be extracted. This method of extracting the enumeration at the centre of symmetry is carried out for maps in [9]. Cycle-pointing and the related constructions provide a general framework to deal with such decompositions and translate them to compact formulas in a systematic way.

5 Random generation

This section provides a general framework to obtain random generators on unlabeled combinatorial classes. We give simple sampling rules associated with each construction, Sum, Product, and Substitution. Then, these rules can be combined to produce in an automatic way a random generator for a class specified from explicit classes using our constructions. The framework we develop is an extension of Boltzmann samplers, as introduced and formalized by Duchon et al. [6].

5.1 The principle of Boltzmann samplers. Let $\widetilde{\mathcal{F}}$ be a family of unlabeled structures ($\mathcal{F}$ is either an unpointed class or a cycle-pointed class). Let $\tilde{f}(x) := \sum_{\gamma \in \widetilde{\mathcal{F}}} x^{|\gamma|}$ be the corresponding ordinary generating function, and ρ the radius of convergence of $\tilde{f}(x)$. A real value $x > 0$ is said to be *admissible* if the sum defining $\tilde{f}(x)$ converges (so $x \leq \rho$). Given a fixed admissible value $x > 0$, a *Boltzmann sampler* $\Gamma\widetilde{\mathcal{F}}(x)$ is a random generator on $\widetilde{\mathcal{F}}$ that draws each structure $\gamma \in \tilde{\mathcal{F}}$ with probability

$$\mathbb{P}(\gamma) = \frac{x^{|\gamma|}}{\tilde{f}(x)}. \tag{5.22}$$

The fundamental property of this probability distribution, called *Boltzmann distribution*, is that two unlabeled structures with the same size have the same probability of being drawn.

As described in [6], there are simple rules to assemble Boltzmann samplers for the two classical constructions Sum and Product (rnd$(0,1)$ stands for a uniform random variable in the interval $(0,1)$):

$\Gamma(\tilde{\mathcal{F}} + \tilde{\mathcal{G}})(x)$: **if** rnd$(0,1) < \frac{\tilde{f}(x)}{\tilde{f}(x)+\tilde{g}(x)}$ **return** $\Gamma\tilde{\mathcal{F}}(x)$
else return $\Gamma\tilde{\mathcal{G}}(x)$
$\Gamma(\tilde{\mathcal{F}} \times \tilde{\mathcal{G}})(x)$: **return** $(\Gamma\tilde{\mathcal{F}}(x), \Gamma\tilde{\mathcal{G}}(x))$

These rules can be used recursively. For instance the class $\mathcal{T}$ of rooted binary trees is specified by $\mathcal{T} = \mathcal{Z} + \mathcal{T} \times \mathcal{T}$, which translates to the Boltzmann sampler

$\Gamma\widetilde{\mathcal{T}}(x)$: **if** rnd$(0,1) < x/T(x)$ **return** leaf
else return $< \Gamma\widetilde{\mathcal{T}}(x), \mathrm{node}, \Gamma\widetilde{\mathcal{T}}(x) >$

Definition. Given an unlabeled class $\widetilde{\mathcal{F}}$ and a fixed tolerance ratio $\epsilon > 0$, an *approximate-size sampler* for $\widetilde{\mathcal{F}}$ is a procedure that, given a target size $n \geq 0$, generates a structure of $\widetilde{\mathcal{F}}$ at random such that the size of the structure generated is in $[n(1-\epsilon), n(1+\epsilon)]$ and the distribution is uniform on each size $k \in [n(1-\epsilon), n(1+\epsilon)]$. An *exact-size sampler* for $\widetilde{\mathcal{F}}$ is a procedure that, given a target size $n \geq 0$, generates at random a structure of $\widetilde{\mathcal{F}}$ of size n, with equal chances for all structures of $\widetilde{\mathcal{F}}_n$.

LEMMA 5.1. (DUCHON ET AL) *Let $\widetilde{\mathcal{F}}$ be an unlabeled combinatorial class, such that the coefficients $\widetilde{f}_n := |\widetilde{\mathcal{F}}_n|$ are asymptotically $\widetilde{f}_n \sim c\rho^{-n}n^{-3/2}$ for some constant c. Assume that there exists a Boltzmann sampler for $\widetilde{\mathcal{F}}$ at $x = \rho$, such that the cost of generating a structure is linearly bounded by the size of the structure all along the generation process. Then, for any fixed $\epsilon > 0$, there exists an approximate size sampler for $\widetilde{\mathcal{F}}$ with expected complexity bounded by $\lambda n/\epsilon$ for some constant λ (i.e., the complexity is linear when ϵ is fixed); and there exists an exact-size sampler for $\widetilde{\mathcal{F}}$ with expected quadratic complexity.*

The approximate-size and exact-size samplers are obtained by running $\Gamma\widetilde{\mathcal{F}}(\rho)$ until the size of the output is in the target domain Ω_n ($\Omega_n = [n(1-\epsilon), n(1+\epsilon)]$ for approximate size sampling and $\Omega_n = \{n\}$ for exact-size sampling). To get the stated complexity, it is necessary that the generation of too large structures is aborted as soon as the size of the generated object gets larger than $\mathrm{Max}(\Omega_n)$.

The asymptotic behaviour in $c\rho^{-n}n^{-3/2}$ is called *universal* [1], as it is very widely encountered in combinatorics (e.g. in all classical families of rooted trees). This asymptotic form occurs in all examples of cycle-pointed structures given in Section 4, except maps.

5.2 Pólya-Boltzmann samplers. Let $\mathcal{A}$ be a combinatorial class (not cycle-pointed). Similarly as for the one-variable case, a vector $(s_i)_{i\geq 1}$ of nonnegative real values is said to be *admissible* if the sum defining $Z_{\mathcal{A}}(s_1, s_2, \ldots)$ converges at $(s_i)_{i\geq 1}$. Given an admissible vector $(s_i)_{i\geq 1}$, a *Pólya-Boltzmann sampler* is a procedure $\Gamma Z_{\mathcal{A}}(s_1, s_2, \ldots)$ that samples symmetries on $\mathcal{A}$ at random such that each symmetry (A, σ) is drawn with probability

$$\mathbb{P}(A, \sigma) = \frac{w_{(A,\sigma)}}{Z_{\mathcal{A}}((s_i)_{i\geq 1})}, \tag{5.23}$$

with $w_{(A,\sigma)}$ as defined in (2.1). Lemma 2.1 ensures that if $s_i = x^i$ and x is admissible for $\tilde{a}(x)$, then $Z_{\mathcal{A}} = \tilde{a}(x)$. In addition, the probability of each symmetry $(A, \sigma) \in \mathcal{S}(\mathcal{A}_n)$ is $x^n/(n!\tilde{a}(x))$. As each unlabeled structure $\tilde{A} \in \tilde{\mathcal{A}}_n$ gives rise to $n!$ symmetries, the procedure of calling $\Gamma Z_{\mathcal{A}}(x, x^2, x^3, \ldots)$, and then returning the underlying unlabeled structure, yields a Boltzmann sampler $\Gamma\tilde{\mathcal{A}}(x)$. Hence, Pólya-Boltzmann samplers are an extension of Boltzmann samplers. This extension makes it possible to have a simple sampling rule for the substitution construction.

Sampling rules:

$\Gamma Z_{\mathcal{A}+\mathcal{B}}(s_i)_{i\geq 1}$
  if $\mathrm{rnd}(0,1) < \frac{Z_{\mathcal{A}}(s_1,s_2,\ldots)}{Z_{\mathcal{A}}(s_1,s_2,\ldots)+Z_{\mathcal{B}}(s_1,s_2,\ldots)}$ **then**
    return $\Gamma Z_{\mathcal{A}}(s_i)_{i\geq 1}$
  else
    return $\Gamma Z_{\mathcal{B}}(s_i)_{i\geq 1}$
  end if

$\Gamma Z_{\mathcal{A}\times\mathcal{B}}(s_i)_{i\geq 1}$
  return $\Gamma Z_{\mathcal{A}}(s_i)_{i\geq 1} \times \Gamma Z_{\mathcal{B}}(s_i)_{i\geq 1}$

$\Gamma Z_{\mathcal{A}\circ\mathcal{B}}(s_i)_{i\geq 1}$
  Compute $(A, \sigma_A) \leftarrow \Gamma Z_{\mathcal{A}}(Z_{\mathcal{B}}^{[i]})_{i\geq 1}$.
  (Recall that $Z_{\mathcal{B}}^{[i]} = Z_{\mathcal{B}}(s_i, s_{2i}, \ldots)$.)
  for each cycle $C = (u_1, \ldots, u_k)$ of σ_A **do**
    Compute $(B, \sigma_B) \leftarrow \Gamma Z_{\mathcal{B}}(s_k, s_{2k} \ldots)$.
    Replace each atom of C by a copy of B
    for each cycle D of σ_B **do**
      Let E be the cycle composed from the copies of D at $u_1, \ldots, u_k$.
    end for
  end for
  return the resulting structure and the automorphism consisting of the composed cycles E

In all these procedures, as well as in the procedures for cycle-pointed classes, the resulting structure S is made well labeled by choosing the atoms from $1, \ldots, |S|$ uniformly at random. The sampling rule for substitution entails the choice of a *starting atom* u_1 in each cycle C of A. This atom is chosen uniformly at random among the atoms of C. In the sampling rule for Product, the product of two symmetries (A, σ_A) and (B, σ_B) is $((A, B), \sigma)$, where σ is the automorphism of (A, B) that consists of the cycles of σ_A and the cycles of σ_B.

Cycle-pointed classes. Given a cycle-pointed class $\mathcal{P}$, a vector $(s_i, t_i)_{i\geq 1}$ of nonnegative real values is said to be *admissible* if the sum of weight-monomials defining $Z_{\mathcal{P}}$ converges when evaluated at this vector. Given a fixed admissible vector $(s_i, t_i)_{i\geq 1}$, a *Pólya-Boltzmann sampler* is a procedure $\Gamma Z_{\mathcal{P}}(s_i, t_i)_{i\geq 1}$ that generates a rooted C-symmetry on $\mathcal{P}$ at random such that each rooted C-symmetry (P, σ, v) of $\mathcal{R}(\mathcal{P})$ is drawn with probability

$$\mathbb{P}(P, \sigma, v) = \frac{w_{(P,\sigma,v)}}{Z_{\mathcal{P}}((s_i, t_i)_{i\geq 1})}, \tag{5.24}$$

with $w_{(P,\sigma,v)}$ as defined in (3.7). Similarly as for classical unlabeled classes, the procedure of calling $\Gamma Z_{\mathcal{P}}(x^i, x^i)_{i\geq 1}$ (with x admissible for the ordinary generating function $\tilde{p}(x)$ of $\tilde{\mathcal{P}}$), and then returning the underlying unlabeled structure of the rooted C-symmetry generated, yields a Boltzmann sampler $\Gamma\tilde{\mathcal{P}}(x)$. The following sampling rules make it possible to assemble Pólya-Boltzmann samplers for cycle-pointed classes.

Sampling rules (cycle-pointed constructions):

$\Gamma Z_{\mathcal{P}+\mathcal{Q}}(s_i,t_i)_{i\geq 1}$

  if $\mathrm{rnd}(0,1) < \frac{Z_P(s_1,t_1;s_2,t_2;\ldots)}{Z_P(s_1,t_1;s_2,t_2;\ldots)+Z_Q(s_1,t_1;s_2,t_2;\ldots)}$
  then
    return $\Gamma Z_{\mathcal{P}}(s_i,t_i)_{i\geq 1}$
  else
    return $\Gamma Z_{\mathcal{Q}}(s_i,t_i)_{i\geq 1}$
  end if

$\Gamma Z_{\mathcal{P}\times\mathcal{A}}(s_i,t_i)_{i\geq 1}$

  return $\Gamma Z_{\mathcal{P}}(s_i,t_i)_{i\geq 1} \times \Gamma Z_{\mathcal{A}}(s_i)_{i\geq 1}$

$\Gamma Z_{\mathcal{A}\times\mathcal{P}}(s_i,t_i)_{i\geq 1}$

  return $\Gamma Z_{\mathcal{A}}(s_i)_{i\geq 1} \times \Gamma Z_{\mathcal{P}}(s_i,t_i)_{i\geq 1}$

$\Gamma Z_{\mathcal{P}\odot(\mathcal{A},\mathcal{Q})}(s_i,t_i)_{i\geq 1}$

  Compute $(P,\sigma_P,v) \leftarrow \Gamma Z_{\mathcal{P}}(Z_{\mathcal{A}}^{[i]}, Z_{\mathcal{Q}}^{[i]})_{i\geq 1}$.
  (Recall that $Z_{\mathcal{Q}}^{[i]} = Z_{\mathcal{Q}}(s_i,t_i;s_{2i},t_{2i};\ldots)$.)
  for each unmarked cycle $C=(u_1,\ldots,u_k)$ of σ_P **do**
    Compute $(A,\sigma_A) \leftarrow \Gamma Z_{\mathcal{A}}(s_k,s_{2k},\ldots)$.
    Replace each atom of C by a copy of A.
    for each cycle D of σ_A **do**
      Let E be the cycle composed from the copies of D at $u_1,\ldots,u_k$.
    end for
  end for
  Let $F=(v_1=v,\ldots,v_l)$ be the marked cycle of P.
  Compute $(Q,\sigma_Q,q) \leftarrow \Gamma Z_{\mathcal{Q}}(s_l,t_l;s_{2l},t_{2l};\ldots)$.
  Replace each atom of F by a copy of Q.
  for each cycle G of σ_Q **do**
    Let H be the cycle composed from the copies of G at $v_1,\ldots,v_l$.
  end for
  In the resulting structure R, mark the cycle composed from the copies of the marked cycle of Q.
  return (R,σ_R,r), where σ_R is the automorphism consisting of the cycles E and the cycles H, and where r is the atom q in the copy of the marked cycle of Q substituted at v.

Note that the evaluation of cycle index sums is only needed for the disjoint union constructions.

Example. Our sampling rules can be used recursively. Consider the class $\mathcal{R}$ of rooted nonplane trees, specified by $\mathcal{R} = \mathcal{Z} \times \textsc{Set} \circ \mathcal{R}$. From the expression $Z_{\textsc{Set}} = \exp(\sum_{i\geq 1} s_i/i)$ it is easy to guess (and then to check) that a Pólya-Boltzmann sampler $\Gamma Z_{\textsc{Set}}(s_i)_{i\geq 1}$ consists in drawing an assembly of cycles with α_i cycles of length i, such that α_i follows a Poisson law $\mathrm{Pois}(s_i/i)$ for $i \geq 1$. This is realized by the following procedure, which operates in linear time (see the complexity model given next):

$\Gamma Z_{\textsc{Set}}(s_i)_{i\geq 1}$

  Draw an integer $k \geq 0$ under the distribution
  $\mathbb{P}(k \leq K) = \exp\left(\sum_{i=1}^{K} \frac{1}{i} s_i\right) / Z_{\textsc{Set}}(s_i)_{i\geq 1}$
  for each $1 \leq i \leq k-1$ **do**
    $\alpha_i \leftarrow \mathrm{Pois}(s_i/i)$
  end for
  $\alpha_k \leftarrow \mathrm{Pois}_{>0}(s_k/k)$ [Poisson conditioned to be not 0]
  return the assembly of cycles with α_i cycles of length i for $1 \leq i \leq k$.

Using the sampling rule for substitution, we obtain a (recursive) Pólya-Boltzmann sampler for $\mathcal{R}$. Via the specialization $(s_i \to x^i)$, this simplifies to the following Boltzmann sampler for $\widetilde{\mathcal{R}}$:

$\Gamma\widetilde{\mathcal{R}}(x)$:

  Draw an integer $k \geq 0$ under the distribution
  $\mathbb{P}(k \leq K) = x \exp\left(\sum_{i=1}^{K} \frac{1}{i} R(x^i)\right) / R(x)$
  for each $1 \leq i \leq k-1$ **do**
    $\alpha_i \leftarrow \mathrm{Pois}(R(x^i)/i)$
    for each $1 \leq j \leq \alpha_i$ **do**
      $T \leftarrow \Gamma\widetilde{R}(x^i)$
      attach i copies of T as children of the root
    end for
  end for
  $\alpha_k \leftarrow \mathrm{Pois}_{>0}(R(x^k)/k)$
  for each $1 \leq j \leq \alpha_k$ **do**
    $T \leftarrow \Gamma\widetilde{R}(x^k)$
    attach k copies of T as children of the root
  end for
  return the obtained rooted tree.

This Boltzmann sampler is also described in [10] using an approach based completely on generating functions (it provides sampling rules for the constructions Multiset and Cycle, but not for substitution).

Oracle-assumption and complexity model. Given a class $\mathcal{F}$ (cycle-pointed or not), we assume that an *oracle* provides the exact value of $Z_{\mathcal{F}}$ at each admissible vector. As in [6], this assumption, called *oracle-assumption*, allows us to separate the discussion on the combinatorial complexity of the algorithms and the complexity related to the evaluations of the cycle index sums. Then the computational cost is defined as the number of combinatorial operations (e.g., creating a vertex or linking two vertices) added to the number of arithmetic operations (comparisons, additions) on real values assumed to be known exactly.

Evaluation-implementation. We frequently use Pólya-Boltzmann samplers as intermediate tools to derive Boltzmann samplers, using $\Gamma Z_{\mathcal{F}}(x,x^2,\ldots) \simeq \Gamma\widetilde{\mathcal{F}}(x)$. Under this specialization, $Z_{\mathcal{F}}(x,x^2,\ldots) = \widetilde{f}(x)$. Hence, we will not need to evaluate cycle-index sums in full generality, but only generating functions, as illustrated in the example of $\Gamma\widetilde{\mathcal{R}}(x)$. In practice, we work with a fixed precision for all real values. For instance, a precision of 20 digits is sufficient to have a negligible bias from uniformity. There exist efficient (recursive) methods to evaluate generating functions such as $R(x)$ with high precision. These techniques also apply to the other classes to be sampled, e.g. the class of cycle-pointed trees, ensuring a small preprocessing cost in each case.

Definition. A class $\mathcal{F}$ is said to have *linear sampling complexity* if there exists a Pólya-Boltzmann sampler $\Gamma Z_{\mathcal{F}}$ for any admissible vector of $\mathcal{F}$, such that the cost of generating a structure is linearly bounded by the size of the structure all along the generation.

Correctness and efficiency of our sampling rules imply the following general theorem, which covers the examples given in Section 4 (see Proposition 5.1 after).

THEOREM 5.1. *Let $\mathcal{F}$ be a combinatorial class recursively specified from classes $\mathcal{G}_1, \ldots, \mathcal{G}_k$ using the constructions $\{+, \times, \circ, \odot\}$, i.e., there exists a decomposition grammar $\{\mathcal{F}_1 = \Psi_1(\mathcal{F}_1, \ldots, \mathcal{F}_m), \ldots, \mathcal{F}_m = \Psi_m(\mathcal{F}_1, \ldots, \mathcal{F}_m)\}$ such that $\mathcal{F} = \mathcal{F}_1$ and the Ψ_i are operations on $\mathcal{F}_1, \ldots, \mathcal{F}_m$ involving the constructions $\{+, \times, \circ, \odot\}$ and the classes $\mathcal{G}_1, \ldots, \mathcal{G}_k$. If the classes $\mathcal{G}_1, \ldots, \mathcal{G}_k$ have linear sampling complexity, then $\mathcal{F}$ has linear sampling complexity as well.*

PROPOSITION 5.1. *Under the oracle assumption, for each of the following* unlabeled *combinatorial classes there exist an approximate-size sampler with expected linear-time complexity and an exact-size sampler with expected quadratic complexity: unrooted (as well as rooted) plane and nonplane trees; any class of unrooted (and rooted) plane and nonplane trees such that the degree of nodes lies in a finite integer set Ω (in particular, acyclic carbon alkanes); RNA secondary structures; connected cactus graphs; connected outerplanar graphs.*

Proof. We give the proof for connected outerplanar graphs. Grammar (4.20) is a decomposition grammar for the class $\mathcal{C}^{\bullet}$ of cycle-pointed outerplanar graphs. The terminal nodes (the classes $\mathcal{G}_1, \ldots, \mathcal{G}_k$) are the classes $\mathcal{Z}, \mathcal{Z}^{\bullet}, \textsc{Set}, \textsc{Set}^{\bullet}_{\geq 2}, \textsc{Set}^{\bullet}, \widehat{\mathcal{D}}, \mathcal{D}^{\bullet}_{\geq 2}, \widehat{\mathcal{D}}^{\bullet}$, which have linear sampling complexity. Hence, Theorem 5.1 ensures that $\mathcal{C}^{\bullet}$ has linear sampling complexity. This gives via $(s_i \to x^i, t_i \to x^i)$ a linear Boltzmann sampler $\Gamma\widetilde{\mathcal{C}}^{\bullet}(x)$. It is proved in [3] that $|\widetilde{\mathcal{C}}_n|$ has the asymptotic form $c\rho^{-n}n^{-5/2}$, so that $|\widetilde{\mathcal{C}}^{\bullet}_n|$ has the universal asymptotic form $c\rho^{-n}n^{-3/2}$ (by $|\widetilde{\mathcal{C}}^{\bullet}_n| = n|\widetilde{\mathcal{C}}_n|$). Hence, Lemma 5.1 ensures that there exist linear approximate-size and quadratic exact-size samplers for $\mathcal{C}^{\bullet}$. These are also approximate-size and exact-size samplers for $\mathcal{C}$, since the pointing operator is *unbiased.*

Acknowledgements. Philippe Flajolet and Omid Amini are greatly thanked for fruitful discussions.

References

[1] J. P. Bell, S. N. Burris, and K. A. Yeats. Counting rooted trees: The universal law $t(n) \sim c\rho^{-n}n^{-3/2}$. http://arxiv.org/abs/math.CO/0512432, 2005.

[2] F. Bergeron, G. Labelle, and P. Leroux. *Combinatorial Species and Tree-like Structures.* Cambridge University Press, 1997.

[3] M. Bodirsky, É. Fusy, M. Kang, and S. Vigerske. Enumeration of unlabeled outerplanar graphs. *Submitted*, 2006.

[4] M. Bodirsky and M. Kang. Generating outerplanar graphs uniformly at random. *Comb. Prob. and Computing*, 15:333–343, 2006.

[5] P. Cameron. *Permutation groups.* Cambridge University Press, 1999.

[6] P. Duchon, P. Flajolet, G. Louchard, and G. Schaeffer. Random sampling from Boltzmann principles. In *ICALP'02*, LNCS, pages 501–513, 2002.

[7] P. Flajolet and R. Sedgewick. *Analytic Combinatorics.* 2006. Book in preparation: available at http://algo.inria.fr/flajolet/Publications/books.html.

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

[9] É. Fusy. Counting unrooted maps using tree-decomposition. In *Proceedings of FPSAC*, 2005.

[10] É. Fusy, P. Flajolet, and C. Pivoteau. Boltzmann sampling of unlabelled structures, 2006. Forthcoming.

[11] F. Harary and E. M. Palmer. *Graphical Enumeration.* Academic Press, New York and London, 1973.

[12] P. Hell and J. Nešetřil. *Graphs and Homomorphisms.* Oxford University Press, 2004.

[13] J. A. Howell, T. F. Smith, and M. S. Waterman. Computation of generating functions for biological molecules. *SIAM J. Appl. Math.*, 39:119–133, 1980.

[14] M. Jerrum. Uniform sampling modulo a group of symmetries using markov chain simulation. Technical Report ECS-LFCS-94-288, Univ. Edinburgh, 1994.

[15] V. Liskovets. A census of non-isomorphic planar maps. In *Coll. Math. Soc. J. Bolyai, Proc. Conf. Algebr. Meth. in Graph Th.*, volume 25:2, pages 479–494, 1981.

[16] A. Nijenhuis and H. Wilf. *Combinatorial algorithms.* Academic Press Inc., 1979.

[17] Otter. The number of trees. *Annals of Math.*, 49:583–599, 1948.

[18] N. Sloane. A000602: Number of n-node unrooted quartic trees. http://www.research.att.com/~njas/sequences/A000602, 2003.

[19] W. T. Tutte. A census of planar maps. *Canad. J. Math.*, 15:249–271, 1963.

[20] H. S. Wilf. The uniform selection of free trees. *J. Algorithms*, 2(2):204–207, 1981.