Fifth Colloquium on Mathematics and Computer Science DMTCS proc. **AI**, 2008, 475–488

Boltzmann Oracle for Combinatorial Systems

Carine Pivoteau[1] and Bruno Salvy[2†]and Michèle Soria[1‡]

[1] *LIP6 – UPMC, 104 avenue du président Kennedy, F-75016 Paris, France,*
[2] *Algorithms Project, Inria Paris-Rocquencourt, France*
`Carine.Pivoteau@lip6.fr, Bruno.Salvy@inria.fr, Michele.Soria@lip6.fr`

Boltzmann random generation applies to well-defined systems of recursive combinatorial equations. It relies on oracles giving values of the enumeration generating series inside their disk of convergence. We show that the combinatorial systems translate into numerical iteration schemes that provide such oracles. In particular, we give a fast oracle based on Newton iteration.

Keywords: Random generation, Boltzmann generation, combinatorics, Newton iteration

1 Introduction

The recent discovery of Boltzmann samplers by Duchon, Flajolet, Louchard and Schaeffer [5] brought a considerable progress to the area of random generation of combinatorial structures. For wide families of classes of structures defined recursively, it is possible to construct automatically efficient random generators (samplers). These generators can produce large structures with the property that a structure of size n is drawn uniformly at random among the c_n structures of size n in its class. They rely on so-called *oracles*, that return numerical values of the generating series of the sequence $(c_n)_{n\geq 0}$ and related series inside their disk of convergence.

In this work, we provide such oracles for numerical values, thus making Boltzmann sampling effective, for all structures specified by well-founded systems of combinatorial equations. The precise setting is given in §2. In the same way that the Boltzmann sampler is constructed automatically from a recursive combinatorial specification, we give automatic constructions of numerical iteration schemes for the oracles. The key point of the treatment is that these numerical schemes are based on combinatorics: each value computed during the numerical iteration is the evaluation of a convergent power series whose coefficients enumerate combinatorial structures.

A straightforward translation of combinatorial systems yields a first family of oracles. For instance, the class of rooted labeled trees is described by the combinatorial equation $\mathcal{T} = \mathcal{Z} \times \textsc{Set}(\mathcal{T})$. This equation can be interpreted as a fixed point equation on combinatorial classes for an appropriate notion of distance (the *contact*). Then the iteration $\mathcal{T}^{[n+1]} = \mathcal{Z} \times \textsc{Set}(\mathcal{T}^{[n]})$ can be seen to converge to the class of rooted labeled trees, starting with $\mathcal{T}^{[0]}$ the empty set. For

† This work is supported in part by the French Research Agency (ANR Gecko).
‡ This work is supported in part by the French Research Agency (ANR Gamma), n°BLAN07-2_195422.

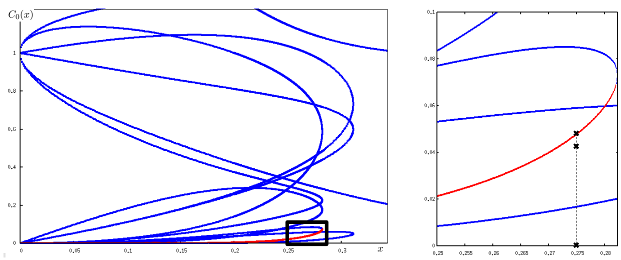
any complex α with $|\alpha| < e^{-1}$, the corresponding numerical iteration $t^{[0]} = 0$, $t^{[n+1]} = \alpha \exp(t^{[n]})$ converges to the value $T(\alpha) = -W(-\alpha)$ of the generating series of rooted labeled trees (W is the Lambert W function). Indeed, each iterate $t^{[n]}$ is equal to the value $T^{[n]}(\alpha)$ of the generating series of the class $\mathcal{T}^{[n]}$ at α. That these series are convergent at α also follows from combinatorics.

Thus our method decomposes into three steps. Starting with a fixed point combinatorial specification (a system of combinatorial equations), we first provide an iteration scheme that converges to the combinatorial solution; then propagate this iteration scheme at the level of generating series and finally at the numerical level, showing that convergence at each stage is a consequence of convergence at the previous one. This is described in §3.

In §4, we design faster oracles based on Newton iteration. For instance, in order to solve the equation $t = \alpha \exp(t)$, Newton's iteration is $t^{[n+1]} = t^{[n]} + (\alpha \exp(t^{[n]}) - t^{[n]})/(1 - \alpha \exp(t^{[n]}))$. Classically, this iteration converges to one of the roots of the equation, provided the starting point is chosen correctly. We show that with $t^{[0]} = 0$, the iteration converges to the root $T(\alpha)$. Moreover, the sequence of iterates is positive increasing so that each iterate is closer to the solution than the previous one. Again, this numerical scheme is based on a *combinatorial* Newton iteration that converges quadratically to the solution. This iteration originates in work by Décoste, Labelle and Leroux [4] (see also [1, ch. 3]), for the computation of species of structures solution to a *univariate* functional equation $\mathcal{Y} = \mathcal{H}(\mathcal{Z}, \mathcal{Y})$. For systems, we have to give a combinatorial meaning to the inverse of the Jacobian matrix. This is handled by the combinatorial "bloomings" introduced by Labelle in his study of the combinatorial Lagrange inversion [10]. Then, as above, this combinatorial iteration transfers to generating series, and quadratic convergence is preserved at the level of valuations. Finally, the numerical version of the iteration yields an oracle in the form of an unconditionally convergent iteration, whose convergence is asymptotically quadratic. Thus, instead of a blind resolution of a system of equations, the iterates directly find their way to the solution that is combinatorially meaningful.

This phenomenon is illustrated in the figure below. A combinatorial structure $\mathcal{C}_0$ is defined by a recursive combinatorial specification. Next, for each value of z in an interval, we plot the solutions (z, C_0) of the system of generating function equations. Only the curve marked in red corresponds to the actual generating function. To the right, we zoom on the black rectangle. The crosses indicate the successive values of the Newton iteration starting from $C_0 = 0$.

$$
\begin{aligned}
\mathcal{C}_0 &= \mathcal{Z}\mathcal{C}_1\mathcal{C}_2\mathcal{C}_3(\mathcal{C}_1 + \mathcal{C}_2)\\
\mathcal{C}_1 &= \mathcal{Z} + \mathcal{Z}\,\mathrm{SEQ}(\mathcal{C}_1^2\mathcal{C}_3^2)\\
\mathcal{C}_2 &= \mathcal{Z} + \mathcal{Z}^2\,\mathrm{SEQ}(\mathcal{Z}\mathcal{C}_2^2\,\mathrm{SEQ}(\mathcal{Z}))\,\mathrm{SEQ}(\mathcal{C}_2)\\
\mathcal{C}_3 &= \mathcal{Z} + \mathcal{Z}(3\mathcal{Z} + \mathcal{Z}^2 + \mathcal{Z}^2\mathcal{C}_1\mathcal{C}_3)\,\mathrm{SEQ}(\mathcal{C}_1^2)
\end{aligned}
$$



Further experiments are described in §5.

In a forthcoming article [13], several extensions are presented: the unlabeled case [8]; specifications defining nonempty structures of size 0; a faster Newton iteration computing much fewer intermediate structures; and a fast algorithm for enumeration sequences.

2 Combinatorial structures

This work uses the combinatorial framework described in detail in the recent book by Flajolet and Sedgewick [6], of which we use the notations and definitions. In this section we recall the basic vocabulary on admissible constructions and specifications that let us express those combinatorial systems we are dealing with. Since not all specifications are combinatorially meaningful, we isolate precisely the class of *well-founded* specifications. These specifications have been considered in the framework of combinatorial classes in [7], and here we show how a combinatorial condition due to Joyal [9] gives an algorithmic characterisation. Finally, we recall what we need from the symbolic method leading to generating series equations.

2.1 Combinatorial specifications

The *admissible constructions* we consider are given in Table 1 (column 1). They form the vocabulary used to express combinatorial specifications. Columns 3 and 4 show how these constructions act on two important special classes: the empty set ($\varnothing$) that does not contain any structure and the neutral class $\mathcal{E}$ consisting of a single structure of size 0.

Definition A *combinatorial specification* for an m-tuple $\boldsymbol{\mathcal{Y}} = (\mathcal{Y}_1, \ldots, \mathcal{Y}_m)$ of classes is a system

$$\begin{cases} \mathcal{Y}_1 = \mathcal{H}_1(\mathcal{Z}, \mathcal{Y}_1, \mathcal{Y}_2, \ldots, \mathcal{Y}_m), \\ \mathcal{Y}_2 = \mathcal{H}_2(\mathcal{Z}, \mathcal{Y}_1, \mathcal{Y}_2, \ldots, \mathcal{Y}_m), \\ \qquad \vdots \\ \mathcal{Y}_m = \mathcal{H}_m(\mathcal{Z}, \mathcal{Y}_1, \mathcal{Y}_2, \ldots, \mathcal{Y}_m), \end{cases}$$

each $\mathcal{H}_i$ denoting a term built from the $\mathcal{Y}_i$'s and the initial class $\mathcal{Z}$ (atomic) using the constructions of Table 1. We refer to this system as $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{H}}(\mathcal{Z}, \boldsymbol{\mathcal{Y}})$; boldfaced letters denote vectors. Each non-terminal $\mathcal{Y}_i$ stands for a combinatorial class and the objects derived by this generalized grammar, starting with axiom $\mathcal{Y}_i$, are called *$\mathcal{Y}_i$-structures*. The *size* of a structure is the number of its terminals $\mathcal{Z}$. The number of elements of size n in a vector of classes $\boldsymbol{\mathcal{Y}}$ is denoted $|\boldsymbol{\mathcal{Y}}|_n$.

construction	notation	$\mathcal{B}=\varnothing$	$\mathcal{B}=\mathcal{E}$	derivative	generating function
Disjoint union	$\mathcal{A}+\mathcal{B}$	$\mathcal{A}$	$\mathcal{A}+\mathcal{E}$	$\mathcal{A}'+\mathcal{B}'$	$A(z)+B(z)$
Cartesian product	$\mathcal{A}\times\mathcal{B}$	$\varnothing$	$\mathcal{A}$	$\mathcal{A}'\times\mathcal{B}+\mathcal{A}\times\mathcal{B}'$	$A(z)\cdot B(z)$
Sequence	$\textsc{Seq}(\mathcal{B})$	$\mathcal{E}$	$-$	$\textsc{Seq}(\mathcal{B})\times\mathcal{B}'\times\textsc{Seq}(\mathcal{B})$	$1/(1-B(z))$
Sequence of card $k>0$	$\textsc{Seq}_k(\mathcal{B})$	$\varnothing$	$-$	$\sum_{i=0}^{k-1}\textsc{Seq}_i(\mathcal{B})\times\mathcal{B}'\times\textsc{Seq}_{k-1-i}(\mathcal{B})$	$B^k(z)$
Cycle	$\textsc{Cyc}(\mathcal{B})$	$\varnothing$	$-$	$\textsc{Seq}(\mathcal{B})\times\mathcal{B}'$	$\log(1/(1-B(z)))$
Cycle of card $k>0$	$\textsc{Cyc}_k(\mathcal{B})$	$\varnothing$	$-$	$\textsc{Seq}_{k-1}(\mathcal{B})\times\mathcal{B}'$	$\frac{1}{k}B^k(z)$
Set	$\textsc{Set}(\mathcal{B})$	$\mathcal{E}$	$-$	$\textsc{Set}(\mathcal{B})\times\mathcal{B}'$	$\exp(B(z))$
Set of card $k>0$	$\textsc{Set}_k(\mathcal{B})$	$\varnothing$	$-$	$\textsc{Set}_{k-1}(\mathcal{B})\times\mathcal{B}'$	$\frac{1}{k!}B^k(z)$

Tab. 1: Admissible constructions, their derivatives and generating functions. The character "−" stands for "undefined". The translation to generating functions of Cycle and Set applies only in the labeled universe, whereas Union, Product and Sequence can be seen in both labeled and unlabeled universes.

2.2 Well-founded combinatorial systems

2.2.1 Definition

We construct oracles for all combinatorial specifications that make sense in the following way.

Definition The combinatorial specification $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{H}}(\mathcal{Z}, \boldsymbol{\mathcal{Y}})$ is *well founded* if and only if, for all $n \geq 0$, it derives only finitely many structures of size n. This is denoted by $|\boldsymbol{\mathcal{Y}}|_n < \infty$.

2.2.2 Derivative

We now introduce an additional admissible construction, which is classical in species theory [1].

Derivative The *derivative* of a term $\mathcal{H}(\mathcal{Z}, \mathcal{Y}_1, \ldots, \mathcal{Y}_m)$ with respect to the non-terminal $\mathcal{Y}_j$ is denoted by $\partial\mathcal{H}/\partial\mathcal{Y}_j$, or $\mathcal{H}'$ when $m = 1$. It is defined recursively, using the rules given in Table 1. The recursion stops at terminals and non-terminals of the term: if $\mathcal{A}$ is either $\mathcal{E}$, $\mathcal{Z}$ or $\mathcal{Y}_k$ with $k \neq j$, then $\partial\mathcal{A}/\partial\mathcal{Y}_j = \varnothing$, while $\partial\mathcal{Y}_j/\partial\mathcal{Y}_j$ is a *new terminal of size 0*, denoted by $\mathcal{Y}_j^\star$.

The only difference with the *pointing* operation considered in [6] is at the end of the recursion, where instead of a new terminal of size 0, the pointing operation is equal to the cartesian product of $\mathcal{Y}_i$ with such a terminal of size 0.

Example. The derivative of the term $\mathcal{H}(\mathcal{Z}, \mathcal{Y}) := \mathcal{Z} \times \textsc{Seq}(\mathcal{Y})$ with respect to $\mathcal{Y}$ is

$$\frac{\partial\mathcal{H}}{\partial\mathcal{Y}} = \frac{\partial\mathcal{Z}}{\partial\mathcal{Y}} \times \textsc{Seq}(\mathcal{Y}) + \mathcal{Z} \times \textsc{Seq}(\mathcal{Y}) \times \frac{\partial\mathcal{Y}}{\partial\mathcal{Y}} \times \textsc{Seq}(\mathcal{Y}) = \mathcal{Z} \times \textsc{Seq}(\mathcal{Y}) \times \mathcal{Y}^\star \times \textsc{Seq}(\mathcal{Y}). \qquad (1)$$

Cartesian product This terminal $\mathcal{Y}_j^\star$ is called a *bud* by Labelle [10]. An induction shows that each non-empty structure of the class $\partial\mathcal{H}/\partial\mathcal{Y}_j(\mathcal{Z}, \mathcal{Y}_1, \ldots, \mathcal{Y}_m)$ contains exactly one bud. This gives a combinatorial interpretation of the cartesian product with a derivative: for any class $\mathcal{C}$, we have the isomorphism

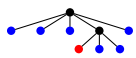
$$\frac{\partial\mathcal{H}}{\partial\mathcal{Y}_j} \times \mathcal{C} \sim \left.\frac{\partial\mathcal{H}}{\partial\mathcal{Y}_j}\right|_{\mathcal{Y}_j^\star = \mathcal{C}}, \qquad (2)$$

where on the right-hand side, we use the substitution of the one bud by $\mathcal{C}$. From now on, we always use this version of the cartesian product with a derivative of one of the $\mathcal{H}_i$'s. The interpretation is that the structures of the cartesian product are obtained by grafting a structure of class $\mathcal{C}$ in the place of a former $\mathcal{Y}_j$ in a structure of $\mathcal{H}$, in all possible ways.

Example. The class $\mathcal{C}$ can be a derivative itself. Thus, the square of the derivative from (1) is

$$\left(\frac{\partial\mathcal{H}}{\partial Y}\right)^2 = \mathcal{Z} \times \textsc{Seq}(\mathcal{Y}) \times (\mathcal{Z} \times \textsc{Seq}(\mathcal{Y}) \times \mathcal{Y}^\star \times \textsc{Seq}(\mathcal{Y})) \times \textsc{Seq}(\mathcal{Y}).$$

A typical structure of this class is depicted below: black dots correspond to locations of a $\mathcal{Z}$, blue dots to the $\mathcal{Y}$'s, the bud $\mathcal{Y}^\star$ is marked in red.

$$(\bullet \bullet \bullet \bullet (\bullet \bullet \bullet \bullet) \bullet) \simeq$$ 

More generally, a sequence $\textsc{Seq}(\frac{\partial H}{\partial Y})$ consists of trees built up by iterating the previous process.

Example. For any term $\mathcal{H}$, the following inclusion holds:

$$\mathcal{H}(\mathcal{Y} + \mathcal{C}) \supset \mathcal{H}(\mathcal{Y}) + \mathcal{H}'(\mathcal{Y}) \times \mathcal{C}. \qquad (3)$$

The interpretation is as follows: the structures on the right-hand side are either in $\mathcal{H}(\mathcal{Y})$, made up only of $\mathcal{Y}$-structures; or in $\mathcal{H}'(\mathcal{Y})\times\mathcal{C}$, structures of $\mathcal{H}(\mathcal{Y})$ with one of the $\mathcal{Y}$-structures replaced by a $\mathcal{C}$-structure. These are clearly disjoint sets, both included in $\mathcal{H}(\mathcal{Y}+\mathcal{C})$. These are the first terms of a complete Taylor formula that has been developed by Labelle [11].

Jacobian matrix As in the classical case, the *Jacobian matrix* of $\boldsymbol{\mathcal{H}}(\mathcal{Z},\boldsymbol{\mathcal{Y}})$ with respect to $\boldsymbol{\mathcal{Y}}$, denoted by $\boldsymbol{\partial\mathcal{H}/\partial\mathcal{Y}}$ is the matrix whose entries are $\partial\mathcal{H}_i(\mathcal{Z},\boldsymbol{\mathcal{Y}})/\partial\mathcal{Y}_j$. The product of a matrix by a vector or by a matrix is obtained by the classical formulas in sums of products forms, sums being interpreted as disjoint unions and products as cartesian products, themselves obtained by grafting at a bud following Eq. (2).
Example. Series-parallel graphs are specified by $(\mathcal{Y}_1,\mathcal{Y}_2)=\boldsymbol{\mathcal{SP}}(\mathcal{Z},\mathcal{Y}_1,\mathcal{Y}_2)$, with

$$\boldsymbol{\mathcal{SP}}(\mathcal{Z},\mathcal{Y}_1,\mathcal{Y}_2)=\begin{pmatrix}\textsc{Seq}_{\geq 2}(\mathcal{Z}+\mathcal{Y}_2)\\ \textsc{Set}_{\geq 2}(\mathcal{Z}+\mathcal{Y}_1)\end{pmatrix},$$

where $\textsc{Seq}_{\geq k}$ is the disjoint union of $\textsc{Seq}_i$ for $i\geq k$, and similarly for $\textsc{Set}_{\geq k}$. Linearity of derivative implies that

$$\frac{\partial\textsc{Seq}_{\geq k}(\mathcal{Y})}{\partial\mathcal{Y}}=\textsc{Seq}_{\geq k-1}(\mathcal{Y})\times\mathcal{Y}^\star\times\textsc{Seq}(\mathcal{Y})+\mathcal{Y}^\star\times\textsc{Seq}_{\geq k-1}(\mathcal{Y}),\quad \frac{\partial\textsc{Set}_{\geq k}(\mathcal{Y})}{\partial\mathcal{Y}}=\textsc{Set}_{\geq k-1}(\mathcal{Y})\times\mathcal{Y}^\star.$$

The Jacobian matrix of our example is therefore

$$\frac{\boldsymbol{\partial\mathcal{SP}}}{\boldsymbol{\partial\mathcal{Y}}}=\begin{pmatrix}\varnothing & \textsc{Seq}_{\geq 1}(\mathcal{Z}+\mathcal{Y}_2)\times\mathcal{Y}_2^\star\times\textsc{Seq}(\mathcal{Z}+\mathcal{Y}_2)+\mathcal{Y}_2^\star\times\textsc{Seq}_{\geq 1}(\mathcal{Z}+\mathcal{Y}_2)\\ \textsc{Set}_{\geq 1}(\mathcal{Z}+\mathcal{Y}_1)\times\mathcal{Y}_1^\star & \varnothing\end{pmatrix}.$$

The Jacobian matrix $\boldsymbol{\partial\mathcal{H}/\partial\mathcal{Y}}$, or any of its powers, is a function of the variables $\mathcal{Z}$ and $\boldsymbol{\mathcal{Y}}$. In particular, evaluating it at $(\mathcal{Z},\boldsymbol{\mathcal{Y}})=(\varnothing,\boldsymbol{\varnothing})$ can be interpreted combinatorially as the matrix whose entry (i,j) is the class of structures obtained by replacing one of the $\mathcal{Y}_j$'s in $\mathcal{H}_i$ by a $\mathcal{Y}_j^\star$, replacing all the other terminals and non-terminals by the empty set and performing the simplifications indicated in Column 3 of Table 1.
Example. The evaluation of the previous matrix at $(\mathcal{Z},\mathcal{Y}_1,\mathcal{Y}_2)=(\varnothing,\varnothing,\varnothing)$ gives $\left(\begin{smallmatrix}\varnothing & \varnothing\\ \varnothing & \varnothing\end{smallmatrix}\right)$.

Nilpotence As in classical linear algebra, a matrix of combinatorial classes is nilpotent if one of its powers is empty (all its entries are the empty set). The order of nilpotence (the minimal power such that emptyness is reached) is bounded by the dimension of the matrix.
Example. For series-parallel graphs, the Jacobian matrix is nilpotent of order 1. An extended system with a third equation $\mathcal{Y}_3=\mathcal{Y}_1+\mathcal{Y}_2$ to define graphs that are either series or parallel has a Jacobian at $(\mathcal{Z},\mathcal{Y}_1,\mathcal{Y}_2,\mathcal{Y}_3)=(\varnothing,\varnothing,\varnothing,\varnothing)$ with third row $\left(\begin{smallmatrix}\mathcal{Y}_1^\star & \mathcal{Y}_2^\star & \varnothing\end{smallmatrix}\right)$. This is nilpotent of order 3.

2.2.3 Characterization of well-founded systems

We can now state an effective criterion for a system to be well founded.

Proposition 1 *A combinatorial specification $\boldsymbol{\mathcal{Y}}=\boldsymbol{\mathcal{H}}(\mathcal{Z},\boldsymbol{\mathcal{Y}})$ such that $\boldsymbol{\mathcal{H}}(\varnothing,\boldsymbol{\varnothing})=\boldsymbol{\varnothing}$ is well founded if and only if the Jocabian matrix $\partial\boldsymbol{\mathcal{H}}/\partial\boldsymbol{\mathcal{Y}}(\varnothing,\boldsymbol{\varnothing})$ is nilpotent.*

Proof: First, we prove by contradiction that nilpotence implies that there is a finite number of $\boldsymbol{\mathcal{Y}}$-structures of any size. Let n be the smallest size for which there is an infinite number of $\boldsymbol{\mathcal{Y}}$-structures. By definition, an arbitrary $\boldsymbol{\mathcal{Y}}$-structure $\boldsymbol{\gamma}$ of size n decomposes as $\boldsymbol{\mathcal{H}}(\mathcal{Z},\gamma_1,\dots,\gamma_m)$, where each γ_i is a $\mathcal{Y}_i$-structure of size at most n. If none of the γ_i has size equal to n, there is a finite number of such decompositions. Otherwise, only one of the γ_i has size n and all the other ones have size 0. The condition $\boldsymbol{\mathcal{H}}(\varnothing, \boldsymbol{\varnothing}) = \boldsymbol{\varnothing}$ implies that the only $\boldsymbol{\mathcal{Y}}$-structure of size 0 is $\boldsymbol{\varnothing}$, thus $\boldsymbol{\gamma}$ is a structure of $\boldsymbol{\partial\mathcal{H}/\partial\mathcal{Y}}(\varnothing, \boldsymbol{\varnothing}) \times \boldsymbol{\beta}$, with $\boldsymbol{\beta}$ a vector of $\boldsymbol{\mathcal{Y}}$-structures of size n. If p is the order of nilpotence of $\boldsymbol{\partial\mathcal{H}/\partial\mathcal{Y}}(\varnothing, \boldsymbol{\varnothing})$, this reasoning cannot be iterated more than p times. Thus there is only a finite number of $\boldsymbol{\mathcal{Y}}$-structures of size n.

Conversely, let $\boldsymbol{\gamma}$ be a $\boldsymbol{\mathcal{Y}}$-structure of size n, with $n = \min\{k,\ |\boldsymbol{\mathcal{Y}}|_k \neq 0\}$. Suppose that the matrix $\boldsymbol{\partial\mathcal{H}/\partial\mathcal{Y}}(\varnothing, \boldsymbol{\varnothing})$ is not nilpotent. Then, for all $q \in \mathbb{N}$, there exists a nonempty structure $\boldsymbol{\beta}_q \in (\boldsymbol{\partial\mathcal{H}/\partial\mathcal{Y}}(\varnothing, \boldsymbol{\varnothing}))^q$ of size 0 so that all $\boldsymbol{\beta}_q \cdot \boldsymbol{\gamma}$ are $\boldsymbol{\mathcal{Y}}$-structures of size n, a contradiction. $\square$

2.3 Generating functions

The *symbolic method* gives a systematic translation of combinatorial constructions into generating functions [6, Thm. II.2]. The basic dictionary is given in Table 1. If a calligraphic letter denotes a class, we use the corresponding roman letter for its generating function.

Test for well-founded specifications To check whether the system is well founded, it is sufficient to: *(i)* translate the combinatorial system into a system over generating series; *(ii)* compute the Jacobian matrix J; *(iii)* compute $J(0,0)^m$, with m the dimension. The system is well founded if and only if this matrix is 0. This improves in simplicity over Zimmermann's algorithm [14].

Definition A combinatorial specification $\boldsymbol{\mathcal{H}}(\mathcal{Z}, \boldsymbol{\mathcal{Y}})$ is called *analytic* when the generating series $\boldsymbol{H}(z, \boldsymbol{Y})$ is analytic in $(z, \boldsymbol{Y})$ in the neighborhood of $(0,\mathbf{0})$, with nonnegative coefficients.

This restriction on combinatorial specifications is necessary for our method. We note that all combinatorial specifications based on the constructions of Table 1 are analytic, by virtue of the analyticity at the origin of $+, \times, \exp, 1/(1-z), \log(1/(1-z))$ and their compositions. Positivity of the coefficients at the origin also follows by induction. Examples of specifications that are *not* analytic in this sense are the constructions of *unlabeled* multiset, powerset or cycles.

3 Iteration and oracle computation

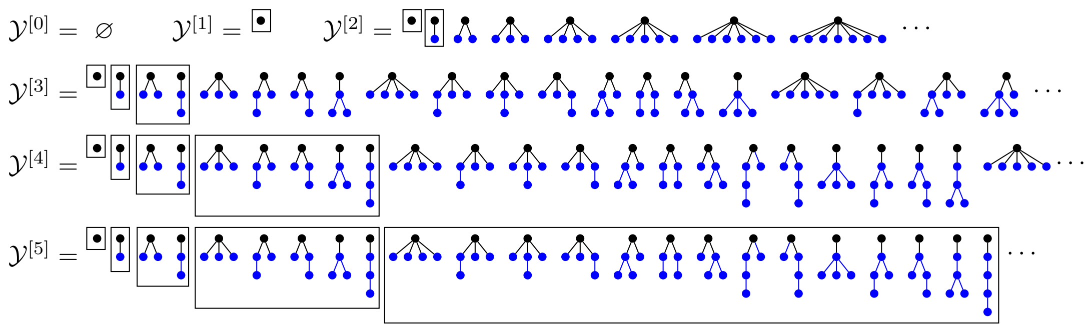
Given a well-founded specification $\boldsymbol{\mathcal{Y}} = \boldsymbol{\Phi}(\mathcal{Z}, \boldsymbol{\mathcal{Y}})$ with the condition $\boldsymbol{\Phi}(\varnothing, \boldsymbol{\varnothing}) = \boldsymbol{\varnothing}$, the natural iteration $\boldsymbol{\mathcal{Y}}^{[n+1]} = \boldsymbol{\Phi}(\mathcal{Z}, \boldsymbol{\mathcal{Y}}^{[n]})$ converges to the solution $\boldsymbol{\mathcal{Y}}$, starting from the vector $\boldsymbol{\mathcal{Y}}^{[0]}$ with empty coordinates (see Theorem 1). In this section, we show that this iteration translates at the level of generating functions, and then at the numerical level when the specification is analytic.

3.1 Example

We first illustrate the processing chain on the simple example of general trees, with specification $\mathcal{Y} = \mathcal{Z} \times \textsc{Seq}(\mathcal{Y})$, so that $\Phi(\mathcal{Z}, \mathcal{Y}) = \mathcal{Z} \times \textsc{Seq}(\mathcal{Y})$.

Check well-founded specification In view of (1), $\Phi(\varnothing, \varnothing) = \varnothing \times \mathcal{E} = \varnothing$ and $\frac{\partial\Phi}{\partial\mathcal{Y}}(\varnothing, \varnothing) = \varnothing$.

Combinatorial iteration The following picture shows the structures produced in the first six iterations of $\mathcal{Y}^{[n+1]} = \mathcal{Z} \times \textsc{Seq}(\mathcal{Y}^{[n]})$.



Rectangles enclose structures when for this size, all structures of the limit $\mathcal{Y}$ have been produced: for example iteration $\mathcal{Y}^{[4]}$ contains all structures of the solution up to size 4; and iteration $\mathcal{Y}^{[5]}$ contains all structures of the solution up to size 5.

Generating series The specification translates into $Y(z) = z/(1 - Y(z))$, while the iteration translates into $Y^{[n+1]}(z) = z/(1 - Y^{[n]}(z))$. The following series are produced by this iteration. By construction, they are the generating series of the classes $\mathcal{Y}^{[0]}, \dots, \mathcal{Y}^{[5]}$ from above. Boldfaced numbers show the coefficients that coincide with those of the series solution $Y(z)$.

$$
\begin{aligned}
&Y^{[0]}(z) = \mathbf{0} \qquad Y^{[1]}(z) = \boldsymbol{z} \\
&Y^{[2]}(z) = \boldsymbol{z} + \boldsymbol{z^2} + z^3 + z^4 + z^5 + z^6 + z^7 + z^8 + z^9 + \cdots \\
&Y^{[3]}(z) = \boldsymbol{z} + \boldsymbol{z^2} + \mathbf{2}\boldsymbol{z^3} + 4z^4 + 8z^5 + 16z^6 + 32z^7 + 64z^8 + 128z^9 + \cdots \\
&Y^{[4]}(z) = \boldsymbol{z} + \boldsymbol{z^2} + \mathbf{2}\boldsymbol{z^3} + \mathbf{5}\boldsymbol{z^4} + 13z^5 + 34z^6 + 89z^7 + 233z^8 + 610z^9 + \cdots \\
&Y^{[5]}(z) = \boldsymbol{z} + \boldsymbol{z^2} + \mathbf{2}\boldsymbol{z^3} + \mathbf{5}\boldsymbol{z^4} + \mathbf{14}\boldsymbol{z^5} + 41z^6 + 122z^7 + 365z^8 + \cdots
\end{aligned}
$$

Numerical values The last step is numerical iteration. In the case of general plane trees, the radius of convergence of the series is $1/4$, so that an evaluation at $\alpha = 0.1$ is possible. The iteration translates into $y^{[n+1]} = 0.1/(1 - y^{[n]})$ and leads to the following values $y^{[i]}$, which again by construction, are the values of the convergent series $Y^{[i]}(z)$, when evaluated at $z = \alpha$, and have for limit $Y(\alpha) = 0.11270166537925831148207346002176\dots$.

$$
\begin{aligned}
&y^{[0]} = \mathbf{0} \qquad y^{[1]} = \mathbf{0.1} \\
&y^{[2]} = \mathbf{0.11}111111111111111111111111111111\dots \\
&y^{[3]} = \mathbf{0.1125}000000000000000000000000000\dots \\
&y^{[4]} = \mathbf{0.1126}7605633802816901408450704225\dots \\
&y^{[5]} = \mathbf{0.1126}9841269841269841269841269841\dots
\end{aligned}
$$

Thus the translation of the combinatorial specification into a numerical iteration provides an oracle for general plane trees. The generating series are not used but their existence and convergence properties play a role in the proof that the numerical iteration converges to the desired value.

3.2 Transfer of Convergence

We now turn to the general case and provide a first oracle based on iterating the specification numerically. We first define the notions of convergence that are needed.

Convergences The notion of contact is classical in species theory [1]. We define its counterpart in the framework of combinatorial classes of [6]. Two combinatorial classes $\mathcal{F}$ and $\mathcal{G}$ have *contact* of order k, denoted by $\mathcal{F} =_k \mathcal{G}$, when their structures of size up to k are identical. A sequence of classes $(\mathcal{Y}^{[n]})_{n\in\mathbb{N}}$ *converges* to a class $\mathcal{Y}$ if for all $k \geq 0$, there exists $N \geq 0$ such that for all $n \geq N$, $\mathcal{Y}^{[n]} =_k \mathcal{Y}$. This is denoted by $\lim_{n\to\infty} \mathcal{Y}^{[n]} = \mathcal{Y}$. Convergence of vectors of combinatorial classes is defined as componentwise convergence.

Recall that the *valuation* of a power series $S(z)$, denoted by $\mathrm{val}(S(z))$, is the exponent of the first nonzero coefficient of the series. A metric is classically deduced by defining the *distance* between two power series by $d(F(z), G(z)) = 2^{-\mathrm{val}(F(z)-G(z))}$; convergence follows.

Theorem 1 (Transfer of Convergence) *Let $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{F}}(\mathcal{Z}, \boldsymbol{\mathcal{Y}})$ be well founded and $\boldsymbol{\mathcal{F}}(\varnothing, \boldsymbol{\varnothing}) = \boldsymbol{\varnothing}$.*

1. *The iteration $\boldsymbol{\mathcal{Y}}^{[n+1]} = \boldsymbol{\mathcal{F}}(\mathcal{Z}, \boldsymbol{\mathcal{Y}}^{[n]})$, with $\boldsymbol{\mathcal{Y}}^{[0]} = \boldsymbol{\varnothing}$, converges to the combinatorial class $\boldsymbol{\mathcal{Y}}$, solution of $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{F}}(\mathcal{Z}, \boldsymbol{\mathcal{Y}})$.*

2. *The iteration $\boldsymbol{Y}^{[n+1]}(z) = \boldsymbol{F}(z, \boldsymbol{Y}^{[n]}(z))$, with $\boldsymbol{Y}^{[0]}(z) = 0$, converges to the generating series $\boldsymbol{Y}(z)$ of the class $\boldsymbol{\mathcal{Y}}$.*

3. *If $\boldsymbol{\mathcal{F}}$ is an analytic specification, then $\boldsymbol{Y}$ has positive radius of convergence ρ and for all α such that $|\alpha| < \rho$, the iteration $\boldsymbol{y}^{[n+1]} = \boldsymbol{F}(\alpha, \boldsymbol{y}^{[n]})$, with $\boldsymbol{y}^{[0]} = 0$, converges to $\boldsymbol{Y}(\alpha)$.*

Proof: 1. Combinatorics The first statement is a consequence of Joyal's proof of his implicit species theorem [9, Thm. 6].

It is useful to note that the sequence $\boldsymbol{\mathcal{Y}}^{[n]}$ is monotonic, in the sense that $\boldsymbol{\mathcal{Y}}^{[n]} \subset \boldsymbol{\mathcal{Y}}^{[n+1]}$ for all n. This is proved by induction. For $n = 0$, this comes from the definition of the class $\boldsymbol{\varnothing}$. Then the inclusion $\boldsymbol{\mathcal{Y}}^{[n]} \subset \boldsymbol{\mathcal{Y}}^{[n+1]}$ is preserved by $\boldsymbol{\mathcal{F}}$ since the structures of $\boldsymbol{\mathcal{F}}(\mathcal{Z}, \boldsymbol{\mathcal{Y}}^{[n]})$ are naturally structures of $\boldsymbol{\mathcal{F}}(\mathcal{Z}, \boldsymbol{\mathcal{Y}}^{[n+1]})$. By definition of the iteration, this means $\boldsymbol{\mathcal{Y}}^{[n+1]} \subset \boldsymbol{\mathcal{Y}}^{[n+2]}$.

2. Power series By the symbolic method, the power series $\boldsymbol{Y}^{[n]}(z)$ are the generating series of the classes $\boldsymbol{\mathcal{Y}}^{[n]}$. Convergence of combinatorial classes translates at the level of generating series into $\mathrm{val}(\boldsymbol{Y}^{[n]}(z) - \boldsymbol{Y}(z)) \to \infty$, which gives the convergence of generating series.

3. Numerical values Since $\boldsymbol{\mathcal{F}}$ is analytic and the Jacobian matrix $(\boldsymbol{I} - \partial\boldsymbol{F}/\partial\boldsymbol{Y})(0, 0)$ is invertible, the implicit function theorem asserts that $\boldsymbol{Y}$ is analytic at 0 (see *e.g.*, [2, Ch. IV]).

The point is to show that for all α such that $0 \leq |\alpha| < \rho$, $\boldsymbol{Y}^{[n]}(\alpha)$ converges to $\boldsymbol{Y}(\alpha)$, and $\boldsymbol{Y}^{[n]}(\alpha) = y^{[n]}$ (the evaluation of $\boldsymbol{Y}^{[n]}$ at α is equal to the value obtained by numerical iteration).

The monotonicity of the combinatorial sequence implies the inequality $[z^k]\boldsymbol{Y}^{[n]}(z) \leq [z^k]\boldsymbol{Y}(z)$ for all n and k. Consequently, the $\boldsymbol{Y}^{[n]}$'s are analytic for $|z| < \rho$, by absolute convergence, and also the tails are bounded by the tail of $\boldsymbol{Y}$. Thus $\boldsymbol{Y}^{[n]}(\alpha)$ converges to $\boldsymbol{Y}(\alpha)$. Let r be such that $|\alpha| \leq r < \rho$. Assuming $\boldsymbol{F}(z, \boldsymbol{Y})$ to be analytic in a polydisc $|(z, \boldsymbol{Y})| \leq (r, \boldsymbol{Y}(r))$

with componentwise inequality, the vector $\boldsymbol{F}(\alpha, \boldsymbol{Y}^{[n]}(\alpha))$ is well defined, and thus by induction $\boldsymbol{y}^{[n+1]} = \boldsymbol{F}(\alpha, \boldsymbol{y}^{[n]}) = \boldsymbol{F}(\alpha, \boldsymbol{Y}^{[n]}(\alpha)) = \boldsymbol{Y}^{[n+1]}(\alpha)$.

We now prove the required analyticity of $\boldsymbol{F}(z, \boldsymbol{Y})$. Let $\boldsymbol{F}(z, \boldsymbol{Y}) = \sum \boldsymbol{f}_{i,\boldsymbol{j}} z^i Y_1^{j_1} \cdots Y_m^{j_m}$ and $\boldsymbol{Y}(z) = \sum \boldsymbol{c}_k z^k$. For each coordinate $h \in \{1, \dots, m\}$, extracting the coefficient of z^k ($k = 0, \dots, N$) in the identity $\boldsymbol{F}(z, \boldsymbol{Y}(z)) = \boldsymbol{Y}(z)$ leads to an inequality of the form

$$\sum_{i+j_1\ell_1+\cdots+j_m\ell_m \le N} f_{h,i,\boldsymbol{j}}\, r^i \Big(\sum_{k_1=0}^{\ell_1} c_{1,k} r^{k_1}\Big)^{j_1} \cdots \Big(\sum_{k_m=0}^{\ell_m} c_{m,k} r^{k_m}\Big)^{j_m} = \sum_{k \le N} c_{h,k} r^k \le Y_h(r), \qquad N \in \mathbb{N},$$

where first indices denote coordinates. The coefficients being positive, the first sum converges to $Y_h(r)$ as $N \to \infty$. This proves the convergence of $F_h(z, \boldsymbol{Y})$ for $|(z, \boldsymbol{Y})| \le (r, \boldsymbol{Y}(r))$ and therefore that of $\boldsymbol{F}(z, \boldsymbol{Y})$ which concludes the proof. $\square$

4 Newton iteration

As the computation for general plane trees in §3.1 exemplifies, the numerical convergence of the previous iteration is typically linear. It is therefore very tempting to use a faster iteration, provided we can show that it converges to the desired solution. The aim of this section is to show that Newton's iteration is appropriate. The proof is based on a combinatorial lifting of Newton's iteration. We first show how this works on the same example.

4.1 Newton Iteration for General Plane Trees

Our oracle for general plane trees is simply the Newton numerical iteration for the equation $y = \alpha/(1-y)$, with initial point 0, *i.e.*,

$$y^{[n+1]} = y^{[n]} + \frac{\alpha/(1-y^{[n]}) - y^{[n]}}{1 - \alpha/(1-y^{[n]})^2}, \qquad y^{[0]} = 0. \tag{4}$$

For $\alpha = 0.1$, the iterates are

$$\begin{aligned}
y^{[0]} &= \mathbf{0}\\
y^{[1]} &= \mathbf{0.11}11111111111111111111111111111\ldots\\
y^{[2]} &= \mathbf{0.112701}25222361359570661896243291 6\ldots\\
y^{[3]} &= \mathbf{0.112701665379 2}3032259476392887392\ldots\\
y^{[4]} &= \mathbf{0.1127016653792583114820734}5989331\ldots
\end{aligned}$$

showing very fast (quadratic) convergence. Note that the equation $y = 0.1/(1-y)$ has another real positive solution, namely $y = 0.88729833462074168852\ldots$. We show below that starting with $y^{[0]} = 0$ ensures convergence to the desired solution.

As in the direct iteration, the proof is based on showing that the values y_i are the evaluations of generating series at 0.1, these generating series enumerating combinatorial classes that are defined by a combinatorial lifting of the Newton iteration.

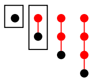
For the case of one equation, such a combinatorial Newton iteration has been introduced by Décoste, Labelle and Leroux [4], who showed that the equation $\mathcal{Y} = \mathcal{F}(\mathcal{Z}, \mathcal{Y})$ is solved by

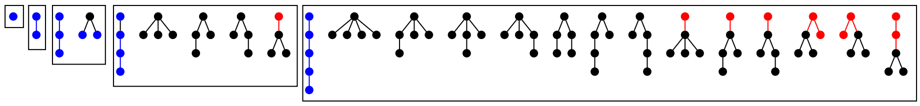
$$\mathcal{Y}^{[n+1]} = \mathcal{Y}^{[n]} + \textsc{Seq}\left(\frac{\partial \mathcal{F}}{\partial \mathcal{Y}}(\mathcal{Y}^{[n]})\right) \times (\mathcal{F}(\mathcal{Z}, \mathcal{Y}^{[n]}) - \mathcal{Y}^{[n]}), \qquad \mathcal{Y}^{[0]} = \varnothing.$$

In the case of general plane trees, $\mathcal{F}(\mathcal{Z}, \mathcal{Y}) = \mathcal{Z} \times \textsc{Seq}(\mathcal{Y})$, so that the iteration becomes

$$\mathcal{Y}^{[n+1]} = \mathcal{Y}^{[n]} + \textsc{Seq}(\mathcal{Z} \times \textsc{Seq}(\mathcal{Y}^{[n]})^2) \times (\mathcal{Z} \times \textsc{Seq}(\mathcal{Y}^{[n]}) - \mathcal{Y}^{[n]}),$$

which should be compared with Eq. (4). The first classes are as follows:

$\mathcal{Y}^{[0]} = \varnothing \qquad \mathcal{Y}^{[1]} =$  $\cdots$

$\mathcal{Y}_{[2]} =$  $\cdots$

The fast convergence can also be seen at the level of generating series:

$$Y^{[0]}(z) = \mathbf{0} \qquad Y^{[1]}(z) = \boldsymbol{z} + \boldsymbol{z^2} + z^3 + z^4 + z^5 + z^6 + z^7 + z^8 + z^9 + \cdots$$
$$Y^{[2]}(z) = \boldsymbol{z} + \boldsymbol{z^2} + \mathbf{2}\boldsymbol{z^3} + \mathbf{5}\boldsymbol{z^4} + \mathbf{14}\boldsymbol{z^5} + \mathbf{42}\boldsymbol{z^6} + 131z^7 + 417z^8 + \cdots$$
$$Y^{[3]}(z) = \boldsymbol{z} + \boldsymbol{z^2} + \mathbf{2}\boldsymbol{z^3} + \mathbf{5}\boldsymbol{z^4} + \mathbf{14}\boldsymbol{z^5} + \mathbf{42}\boldsymbol{z^6} + \cdots + \mathbf{742900}\boldsymbol{z^{14}} + \cdots$$
$$Y^{[4]}(z) = \boldsymbol{z} + \boldsymbol{z^2} + \mathbf{2}\boldsymbol{z^3} + \mathbf{5}\boldsymbol{z^4} + \mathbf{14}\boldsymbol{z^5} + \mathbf{42}\boldsymbol{z^6} + \cdots + \mathbf{1002242216651368}\boldsymbol{z^{30}} + \cdots$$

4.2 General Case

Our main result is the following.

Theorem 2 (Newton Oracle) *Let $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{H}}(\mathcal{Z}, \boldsymbol{\mathcal{Y}})$ be a well-founded analytic specification with $\boldsymbol{\mathcal{H}}(\varnothing, \boldsymbol{\varnothing}) = \boldsymbol{\varnothing}$. Let α be inside the disk of convergence of the generating series $\boldsymbol{Y}(z)$ of $\boldsymbol{\mathcal{Y}}$. Then the following iteration converges to $\boldsymbol{Y}(\alpha)$:*

$$\boldsymbol{y}^{[n+1]} = \boldsymbol{y}^{[n]} + \left(\mathbf{I} - \frac{\partial \boldsymbol{H}}{\partial \boldsymbol{Y}}(\alpha, \boldsymbol{y}^{[n]})\right)^{-1} \cdot (\boldsymbol{H}(\alpha, \boldsymbol{y}^{[n]}) - \boldsymbol{y}^{[n]}), \qquad \boldsymbol{y}^{[0]} = \mathbf{0}.$$

This is the classical Newton iteration for systems. The main point here is that given the initial point $\mathbf{0}$, not only does the iteration converge, but the limit is the desired value. Our proof relies on an extension of the Newton combinatorial iteration [4] to the case of systems.

Proposition 2 *Let $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{H}}(\mathcal{Z}, \boldsymbol{\mathcal{Y}})$ be a well-founded specification with $\boldsymbol{\mathcal{H}}(\varnothing, \boldsymbol{\varnothing}) = \boldsymbol{\varnothing}$. Let $\boldsymbol{\mathcal{N}}_{\boldsymbol{\mathcal{H}}}$ be the operator defined by*

$$\boldsymbol{\mathcal{N}}_{\boldsymbol{\mathcal{H}}}(\mathcal{Z}, \boldsymbol{\mathcal{Y}}) = \boldsymbol{\mathcal{Y}} + \left(\mathbf{I} - \frac{\partial \boldsymbol{\mathcal{H}}}{\partial \boldsymbol{\mathcal{Y}}}(\mathcal{Z}, \boldsymbol{\mathcal{Y}})\right)^{-1} \times (\boldsymbol{\mathcal{H}}(\mathcal{Z}, \boldsymbol{\mathcal{Y}}) - \boldsymbol{\mathcal{Y}}).$$

Then the sequence defined by $\boldsymbol{\mathcal{Y}}^{[0]} = \boldsymbol{\varnothing}$, $\boldsymbol{\mathcal{Y}}^{[n+1]} = \boldsymbol{\mathcal{N}}_{\boldsymbol{\mathcal{H}}}(\mathcal{Z}, \boldsymbol{\mathcal{Y}}^{[n]})$ $(n \geq 0)$ converges to $\boldsymbol{\mathcal{Y}}$. Moreover this convergence is quadratic: $\boldsymbol{\mathcal{Y}}^{[n]} =_k \boldsymbol{\mathcal{Y}} \Rightarrow \boldsymbol{\mathcal{Y}}^{[n+1]} =_{2k+1} \boldsymbol{\mathcal{Y}}$.

Concerning the combinatorial meaning of the iteration, a few words of explanation are in order. Following Labelle [10], the inverse of the Jacobian matrix is given by:

$$\left(\mathbf{I} - \frac{\partial \boldsymbol{\mathcal{H}}}{\partial \boldsymbol{\mathcal{Y}}}(\mathcal{Z}, \boldsymbol{\mathcal{Y}})\right)^{-1} = \sum_{k \geq 0} \left(\frac{\partial \boldsymbol{\mathcal{H}}}{\partial \boldsymbol{\mathcal{Y}}}(\mathcal{Z}, \boldsymbol{\mathcal{Y}})\right)^{k}.$$

The subtraction $\boldsymbol{\mathcal{H}}(\mathcal{Z}, \boldsymbol{\mathcal{Y}}) - \boldsymbol{\mathcal{Y}}$ is defined when $\boldsymbol{\mathcal{Y}} \subset \boldsymbol{\mathcal{H}}(\mathcal{Z}, \boldsymbol{\mathcal{Y}})$ as the set difference of the classes.

Example. For series-parallel graphs, starting with $\boldsymbol{\mathcal{Y}}^{[0]} = (\varnothing, \varnothing)$, the first step of the combinatorial Newton iteration produces the combinatorial class

$$\boldsymbol{\mathcal{Y}}^{[1]} = \sum_{k \geq 0} \begin{pmatrix} \varnothing & \textsc{Seq}_{\geq 1}(\mathcal{Z}) \times \mathcal{Y}_2^{\star} \times \textsc{Seq}(\mathcal{Z}) + \mathcal{Y}_2^{\star} \times \textsc{Seq}_{\geq 1}(\mathcal{Z}) \\ \textsc{Set}_{\geq 1}(\mathcal{Z}) \times \mathcal{Y}_1^{\star} & \varnothing \end{pmatrix}^{k} \cdot \begin{pmatrix} \textsc{Seq}_{\geq 2}(\mathcal{Z}) \\ \textsc{Set}_{\geq 2}(\mathcal{Z}) \end{pmatrix}.$$

For arbitrary n, the *numerical* Newton iteration is

$$\boldsymbol{y}^{[n+1]} = \boldsymbol{y}^{[n]} + \begin{pmatrix} 1 & 1 - \frac{1}{(1-\alpha-y_2^{[n]})^2} \\ 1 - e^{\alpha + y_1^{[n]}} & 1 \end{pmatrix}^{-1} \cdot \begin{pmatrix} \frac{(\alpha+y_2^{[n]})^2}{1-\alpha-y_2^{[n]}} - y_1^{[n]} \\ e^{\alpha + y_1^{[n]}} - 1 - \alpha - y_1^{[n]} - y_2^{[n]} \end{pmatrix}.$$

The singularity is at $2 - \sqrt{5} + \ln((1+\sqrt{5})/2) \approx 0.245$. With $\alpha = 0.24$, the first few iterates are

$$\begin{aligned}
\boldsymbol{y}^{[1]} &= (\mathbf{0.1}2305106632099430637 22\ldots, && \mathbf{0.0}646266475071172143 9535\ldots\ )\\
\boldsymbol{y}^{[2]} &= (\mathbf{0.1}6270003893196157969 26\ldots, && \mathbf{0.09}2012932660348777349 70\ldots\ )\\
\boldsymbol{y}^{[3]} &= (\mathbf{0.17}2433330700324571068 6\ldots, && \mathbf{0.097}984418035783383360 38\ldots\ )\\
\boldsymbol{y}^{[4]} &= (\mathbf{0.17304}6096550753535357 4\ldots, && \mathbf{0.09836}831514307466499845\ldots\ )\\
\boldsymbol{y}^{[5]} &= (\mathbf{0.1730486392}973095133433\ldots, && \mathbf{0.0983698991}7963665326450\ldots\ )\\
\boldsymbol{y}^{[6]} &= (\mathbf{0.1730486393408452105}149\ldots, && \mathbf{0.0983698992067876912 6}015\ldots)
\end{aligned}$$

Proof of Proposition 2: The proof is based on the same three steps as that of the combinatorial Newton iteration in the univariate case [4].

1. The iteration is well defined The subtraction is possible only if $\boldsymbol{\mathcal{Y}}^{[n]} \subset \boldsymbol{\mathcal{H}}(\mathcal{Z}, \boldsymbol{\mathcal{Y}}^{[n]})$. The proof of this inclusion is by induction. For $n = 0$ this is a consequence of $\boldsymbol{\mathcal{Y}}^{[0]}$ being the empty set. If the property is satisfied for n, then we use Eq. (3) with $\mathcal{A} = \boldsymbol{\mathcal{Y}}^{[n]}$ and $\mathcal{B} = \boldsymbol{\mathcal{Y}}^{[n+1]} - \boldsymbol{\mathcal{Y}}^{[n]}$. This latter subtraction is justified since the first summand of $\boldsymbol{\mathcal{N}}_{\boldsymbol{\mathcal{H}}}(\mathcal{Z}, \boldsymbol{\mathcal{Y}}^{[n]})$ is $\boldsymbol{\mathcal{Y}}^{[n]}$. Thus we get

$$\begin{aligned}
\boldsymbol{\mathcal{H}}(\mathcal{Z}, \boldsymbol{\mathcal{Y}}^{[n+1]}) &\supset \boldsymbol{\mathcal{H}}(\mathcal{Z}, \boldsymbol{\mathcal{Y}}^{[n]}) + \frac{\partial \boldsymbol{\mathcal{H}}}{\partial \boldsymbol{\mathcal{Y}}}(\mathcal{Z}, \boldsymbol{\mathcal{Y}}^{[n]}) \times (\boldsymbol{\mathcal{Y}}^{[n+1]} - \boldsymbol{\mathcal{Y}}^{[n]}),\\
&\supset \boldsymbol{\mathcal{Y}}^{[n]} + (\boldsymbol{\mathcal{H}}(\mathcal{Z}, \boldsymbol{\mathcal{Y}}^{[n]}) - \boldsymbol{\mathcal{Y}}^{[n]})\\
&\qquad + \frac{\partial \boldsymbol{\mathcal{H}}}{\partial \boldsymbol{\mathcal{Y}}}(\mathcal{Z}, \boldsymbol{\mathcal{Y}}^{[n]}) \times \sum_{k \geq 0} \left(\frac{\partial \boldsymbol{\mathcal{H}}}{\partial \boldsymbol{\mathcal{Y}}}(\mathcal{Z}, \boldsymbol{\mathcal{Y}}^{[\boldsymbol{n}]})\right)^{k} \times (\boldsymbol{\mathcal{H}}(\mathcal{Z}, \boldsymbol{\mathcal{Y}}^{[n]}) - \boldsymbol{\mathcal{Y}}^{[n]}),\\
&\supset \boldsymbol{\mathcal{Y}}^{[n]} + \sum_{k \geq 0} \left(\frac{\partial \boldsymbol{\mathcal{H}}}{\partial \boldsymbol{\mathcal{Y}}}(\mathcal{Z}, \boldsymbol{\mathcal{Y}}^{[\boldsymbol{n}]})\right)^{k} \times (\boldsymbol{\mathcal{H}}(\mathcal{Z}, \boldsymbol{\mathcal{Y}}^{[n]}) - \boldsymbol{\mathcal{Y}}^{[n]}) = \boldsymbol{\mathcal{Y}}^{[n+1]},
\end{aligned}$$

where in the second line we use the induction hypothesis to rewrite $\mathcal{H}(\mathcal{Z},\mathcal{Y}^{[n]})$ and the definition of the iteration to rewrite $\mathcal{Y}^{[n+1]}$.

2. The iteration is not ambiguous All the structures of $\mathcal{N}_{\mathcal{H}}(\mathcal{Z},\mathcal{Y}^{[n]})$ are distinct: this comes from an induction using the fact that the final grafting of an element of $\mathcal{H}(\mathcal{Z},\mathcal{Y}^{[n]})-\mathcal{Y}^{[n]}$ cannot occur anywhere else in a structure built on $\mathcal{Y}^{[n]}$'s only.

3. Convergence and its quadratic behaviour Assume that $\mathcal{Y}^{[n-1]} =_k \mathcal{Y}^{[n]}$ and consider an element γ of size at most $2k+1$ in the class $\mathcal{Y}^{[n+1]}-\mathcal{Y}^{[n]}$. Thus γ belongs to $(\partial\mathcal{H}/\partial\mathcal{Y}(\mathcal{Z},\mathcal{Y}^{[n]}))^{\ell}\times(\mathcal{H}(\mathcal{Z},\mathcal{Y}^{[n]})-\mathcal{Y}^{[n]})$ for some $\ell\in\mathbb{N}$. Exactly one of the elements of $\mathcal{Y}^{[n]}$ composing γ has size larger than k: two of them would give γ a size larger than $2k+1$ and none of them would make all these elements of $\mathcal{Y}^{[n]}$ elements of $\mathcal{Y}^{[n-1]}$ and therefore γ would belong to $\mathcal{Y}^{[n]}$. Moreover, this element, say $\boldsymbol{\beta}$, belongs to the final $\mathcal{H}(\mathcal{Z},\mathcal{Y}^{[n]})$ since otherwise this final element, say $\boldsymbol{\delta}$, would belong to $\mathcal{H}(\mathcal{Z},\mathcal{Y}^{[n-1]})\subset\mathcal{Y}^{[n]}$. By definition, $\boldsymbol{\beta}$ itself belongs to $(\partial\mathcal{H}/\partial\mathcal{Y}(\mathcal{Z},\mathcal{Y}^{[n-1]}))^{\ell'}\times(\mathcal{H}(\mathcal{Z},\mathcal{Y}^{[n-1]})-\mathcal{Y}^{[n-1]})$ for some $\ell'\in\mathbb{N}$. Thus $\boldsymbol{\delta}$ belongs to $(\partial\mathcal{H}/\partial\mathcal{Y}(\mathcal{Z},\mathcal{Y}^{[n-1]}))^{\ell'+1}(\mathcal{H}(\mathcal{Z},\mathcal{Y}^{[n-1]})-\mathcal{Y}^{[n-1]})$ which is part of $\mathcal{Y}^{[n]}$, and that is not possible. Thus there is no γ of size at most $2k+1$ in $\mathcal{Y}^{[n+1]}-\mathcal{Y}^{[n]}$, or equivalently $\mathcal{Y}^{[n+1]} =_{2k+1} \mathcal{Y}^{[n]}$.

The proof is concluded by showing that the limit is the solution of $\mathcal{Y}=\mathcal{H}(\mathcal{Z},\mathcal{Y})$. This follows from rewriting the iteration in the form

$$\mathcal{H}(\mathcal{Z},\mathcal{Y}^{[n]})-\mathcal{Y}^{[n]}+\frac{\partial\mathcal{H}}{\partial\mathcal{Y}}(\mathcal{Z},\mathcal{Y}^{[n]})\cdot(\mathcal{Y}^{[n+1]}-\mathcal{Y}^{[n]})=\mathcal{Y}^{[n+1]}-\mathcal{Y}^{[n]}$$

and observing that since $\mathcal{Y}^{[n+1]}-\mathcal{Y}^{[n]}$ converges to $\varnothing$, so does $\mathcal{H}(\mathcal{Z},\mathcal{Y}^{[n]})-\mathcal{Y}^{[n]}$. □

Proof of Theorem 2: Since the limit of $\mathcal{Y}^{[n+1]}=\mathcal{N}_{\mathcal{H}}(\mathcal{Z},\mathcal{Y}^{[n]})$ is the solution of $\mathcal{Y}=\mathcal{H}(\mathcal{Z},\mathcal{Y})$, which is well founded, there are only finitely many elements in $\mathcal{Y}^{[n]}$ of any size in $\mathcal{Y}$ and this makes the specification $\mathcal{Y}=\mathcal{N}_{\mathcal{H}}(\mathcal{Z},\mathcal{Y})$ well founded too. It is analytic by the analyticity of $\mathcal{H}$. The proof is completed by application of part (3) of Theorem 1 with $\mathcal{F}=\mathcal{N}_{\mathcal{H}}$. □

5 Applications

Figure 1 displays the timings of a straightforward Maple implementation of the Newton oracle for our example of series-parallel graphs. Each point corresponds to a computation of the oracle at a specified α and precision(i). The curves display jumps corresponding to precisions where the number of Newton iterations is increased by one.

We have also implemented a prototype of an optimized Newton iteration [13]. Our prototype first decomposes the grammar into strongly connected components and each Newton iteration is run on these components separately. Indications on the performance are given in Table 2. In these tests, we generated random grammars for different values of the number of equations and the number of constructions per equation. The average size of the largest strongly connected component of the system is given in the third row, since it has a significant impact on the performance. Timings are given in seconds. In applications to Boltzmann generation, the location

(i) The program was run using Maple 11 on an Intel processor at 3.2 GHz with 2 GB of memory.

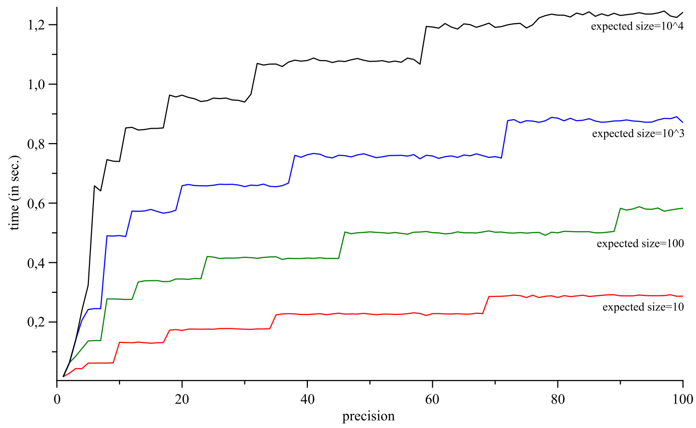


Fig. 1: Evaluation of the Newton Oracle for series-parallel graphs (time vs precision). Expected sizes 10, 100, 1,000, 10,000 (corresponding to $\alpha \simeq 0.2290606680; 0.2450366213; 0.2451428138; 0.2451438373$).

of the point α at which the oracle is invoked is related to the expected size of the objects produced by the random generator, which increases with the closeness to the dominant singularity. Thus, we give our timings at points on a scale given by the dominant singularity ρ. In the last row, we give the average value of the expected size of the objects produced by the Boltzmann generators at 0.999999ρ, where the average is taken over 100 grammars in each column.

# equations	4	10	50		100		500
# constructions/eqn	10	10	10	50	10	50	50
avg size largest scc	2.47	3.42	7.95	18.62	10.93	67.18	339.1
time (0.99ρ)	0.05	0.11	0.17	0.47	0.23	7.29	61.73
time (0.999999ρ)	0.08	0.16	0.19	0.56	0.25	8.11	61.86
avg expected size	$4.1\,10^{14}$	$1.4\,10^{7}$	$2.2\,10^{5}$	$1.0\,10^{5}$	$1.2\,10^{6}$	$5.0\,10^{4}$	$3.3\,10^{4}$

Tab. 2: Experimental Results

Besides these random examples, our prototype has also been used to compute oracles and generate random structures for systems appearing in real applications. Darrasse has applied the method to generate XML documents [3]: he developed a program that takes any RELAX NG grammar and produces huge uniform random trees that are valid XML documents. These documents can be used, for instance, for testing vulnerability of web services. With context-free grammars such as MathML or DocBook, leading to combinatorial systems of hundreds of equations, he could compute the oracles in a few seconds, and produce random documents of 10,000 nodes in less than one minute on a standard PC.

In a preliminary work with Oudinet, concerned with paths in large graphs modeling concurrent systems [12], we could deal with a first example comprising 1,183 equations in a few seconds, at a point which leads to an expected size of approximately 100,000 nodes.

References

[1] F. Bergeron, G. Labelle, and P. Leroux. *Combinatorial species and tree-like structures*, volume 67 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, 1998. ISBN 0-521-57323-8. Translated from the 1994 French original.

[2] H. Cartan. *Elementary theory of analytic functions of one or several complex variables*. Dover Publications, 1995. ISBN 0486685438 (pbk.). URL `http://www.loc.gov/catdir/description/dover031/95013507.html`. Reprint of the 1973 ed, translated from the 1961 French original.

[3] A. Darrasse. Random XML sampling the Boltzmann way. Technical Report 0807.0992, arXiv, 2008. URL `http://arxiv.org/abs/0807.0992`.

[4] H. Décoste, G. Labelle, and P. Leroux. Une approche combinatoire pour l'itération de Newton-Raphson. *Advances in Applied Mathematics*, 3:407–416, 1982.

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

[6] P. Flajolet and R. Sedgewick. *Analytic Combinatorics*. Cambridge University Press, 2008. To be published, available from P. Flajolet's web page.

[7] P. Flajolet, B. Salvy, and P. Zimmermann. Automatic average-case analysis of algorithms. *Theoretical Computer Science Series A*, 79(1):37–109, Feb. 1991.

[8] P. Flajolet, É. Fusy, and C. Pivoteau. Boltzmann sampling of unlabelled structures. In D. Applegate, G. S. Brodal, D. Panario, and R. Sedgewick, editors, *Proceedings of the Ninth Workshop on Algorithm Engineering and Experiments and the Fourth Workshop on Analytic Algorithmics and Combinatorics*, volume 126 of *SIAM Proceedings in Applied Mathematics*, pages 201–211. SIAM, 2007. Workshops held in New Orleans, LA, January 2007.

[9] A. Joyal. Une théorie combinatoire des séries formelles. *Advances in Mathematics*, 42(1): 1–82, 1981.

[10] G. Labelle. Éclosions combinatoires appliquées à l'inversion multidimensionnelle des séries formelles. *Journal of Combinatorial Theory. Series A*, 39(1):52–82, 1985. ISSN 0097-3165.

[11] G. Labelle. Dérivées directionnelles et développements de Taylor combinatoires. *Discrete Mathematics*, 79(3):279–297, 1990. ISSN 0012-365X.

[12] J. Oudinet. Uniform random walks in very large models. In M.-C. Gaudel, J. Mayer, and R. Merkel, editors, *Proceedings of the Second International Workshop on Random Testing (RT 2007)*, pages 26–29, Atlanta, GA, USA, November 2007. ACM Press.

[13] C. Pivoteau, B. Salvy, and M. Soria. Newton iteration for combinatorial systems with applications to enumeration and random generation. In preparation, 2008.

[14] P. Zimmermann. *Séries génératrices et analyse automatique d'algorithmes*. PhD thesis, École polytechnique, Palaiseau, France, 1991.