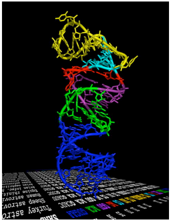
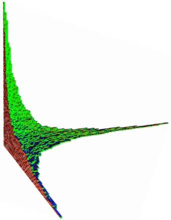
Boltzmann sampling

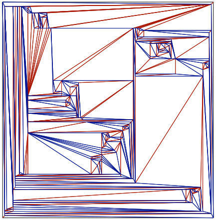
Carine Pivoteau

LIP6 – UPMC

based on work by P. Duchon, P. Flajolet, E. Fusy, G. Louchard, C. Pivoteau and G. Schaeffer







Outline of the talk

1. Introduction
2. Boltzmann model and free samplers
3. Effective samplers

Random generation: different approaches

Fixed size random uniform generation:

- Ad hoc methods
  - bijections, surjections, ...
    $\mathcal{A} = \phi(\mathcal{B})$ and $\Gamma\mathcal{B}(n) \Rightarrow$ random sampler $\Gamma\mathcal{A}(n)$
    $a_n = f(a_{n-1}) \Rightarrow$ incremental algorithm $\Gamma\mathcal{A}(n)$
  - rejection
    $\mathcal{A} \subset \mathcal{B}$ and $\Gamma\mathcal{B}(n) \Rightarrow$ random sampler $\Gamma\mathcal{A}(n)$
- Recursive method : counting + recursive process
  - *Nijenhuis, Wilf, 1978*
  - *Flajolet, Zimmermann, Van Cutsem, 1994*
    preprocessing time (to compute g.f. coefficients): $O(n^2)$
    random generation time : $O(n \log n)$

Approximate size random uniform generation:

- Boltzmann sampling...

Constructible classes [Flajolet, Sedgewick]

- decomposable combinatorial structures
- grammar : $\mathcal{E}$, $\mathcal{Z}$, $+$, $\times$, sequence, cycle, set (labelled or unlabelled)

$\textsc{Set}(\textsc{Seq}(\mathcal{Z}, \# \geq 1))$	integer partitions	unlabelled
$\text{P}\textsc{Set}(\textsc{Seq}(\mathcal{Z}, \# \geq 1))$	integer partitions without repetition	unlabelled
$\begin{cases} \mathcal{S} = \textsc{Seq}_{\geq 2}(\mathcal{P} + \mathcal{Z}) \\ \mathcal{P} = \textsc{Set}_{\geq 2}(\mathcal{S} + \mathcal{Z}) \end{cases}$	series-parallel graphs	labelled
$\mathcal{B} = \mathcal{Z} + \mathcal{B} \times \mathcal{B}$	plane binary trees	(un)labelled
$\mathcal{T} = \mathcal{Z} \times \text{P}\textsc{Set}(\mathcal{T}))$	general nonplane trees	(un)labelled
$\begin{cases} \mathcal{G} = \text{M}\textsc{Set}(\textsc{Cyc}(\mathcal{T})) \\ \mathcal{T} = \mathcal{Z} \times \text{M}\textsc{Set}(\mathcal{T})) \end{cases}$	functional graphs	(un)labelled

- size function
- automatic generating functions

g.f. of a combinatorial class $\mathcal{C}$: $C(z) = \sum_{n \geq 0} c_n z^n$ $\qquad \hat{C}(z) = \sum_{n \geq 0} c_n \frac{z^n}{n!}$

where c_n is the number of objects of $\mathcal{C}$ which have size n.

Constructible classes – summary

	specification	ordinary g.f. (unlabelled)	exponential g.f. (labelled)
ε / atom	1 / $\mathcal{Z}$	1 / x	1 / x
Union	$\mathcal{C} = \mathcal{A} \cup \mathcal{B}$	$C(x) = A(x) + B(x)$	$\hat{C}(x) = A(x) + B(x)$
Product	$\mathcal{C} = \mathcal{A} \times \mathcal{B}$	$C(x) = A(x) \times B(x)$	$\hat{C}(x) = A(x) \times B(x)$
Sequence	$\mathcal{C} = \textsc{Seq}(\mathcal{A})$	$C(x) = \frac{1}{1-A(x)}$	$\hat{C}(x) = \frac{1}{1-A(x)}$
PowerSet	$\mathcal{C} = \textsc{PSet}(\mathcal{A})$	$\exp\left(\sum_{k=1}^{\infty} \frac{(-1)^{k-1}}{k} A(x^k)\right)$	$\hat{C}(x) = \exp(A(x))$
Multiset	$\mathcal{C} = \textsc{MSet}(\mathcal{A})$	$\exp\left(\sum_{k=1}^{\infty} \frac{1}{k} A(x^k)\right)$	–
Cycle	$\mathcal{C} = \textsc{Cyc}(\mathcal{A})$	$\sum_{k=1}^{\infty} \frac{\varphi(k)}{k} \log \frac{1}{1-A(x^k)}$	$\hat{C}(x) = \log \frac{1}{1-A(x)}$

Boltzmann model and free samplers

Boltzmann method

Random sampling under Boltzmann model

- approximate size sampling,
- size distribution spread over the whole combinatorial class, but uniform for a sub-class of objects of the same size,
- control parameter,
- automatized sampling: the sampler is compiled from specification automatically,
- very large objects can be sampled.
  - → large scale simulations
  - → observation of random structures limit properties...

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

Boltzmann sampling of unlabelled structures. Ph. Flajolet, E. Fusy, C. Pivoteau. Proceedings of ANALCO07, january 2007.

Model definition

Definition

In the unlabelled case, Boltzmann model assigns to any object $c \in \mathcal{C}$ the following probability:

$$\mathbb{P}_x(c) = \frac{x^{|c|}}{C(x)}$$

In the labelled case, this probability becomes:

$$\mathbb{P}_x(c) = \frac{1}{\hat{C}(x)} \frac{x^{|c|}}{|c|!}$$

A free Boltzmann sampler $\Gamma C(x)$ for the class $\mathcal{C}$ is a process that produces objects from $\mathcal{C}$ according to this model.

→ 2 objects of the same size will be drawn with the same probability.

1 Introduction

2 Boltzmann model and free samplers
- Basic constructions
- Labelled sets ans cycles
- Back to unlabelled

3 Effective samplers

Unlabelled unions, products, sequences

Suppose $\Gamma A(x)$ and $\Gamma B(x)$ are given:

Disjoint unions

Boltzmann sampler ΓC for $\mathcal{C} = \mathcal{A} \cup \mathcal{B}$:

With probability $\frac{A(x)}{C(x)}$ do $\Gamma A(x)$ else do $\Gamma B(x)$ $\rightarrow$ Bernoulli.

Products

Boltzmann sampler ΓC for $\mathcal{C} = \mathcal{A} \times \mathcal{B}$:

Generate a pair $\langle\ \Gamma A(x)\ ,\ \Gamma B(x)\ \rangle$ $\rightarrow$ independent calls.

Sequences

Boltzmann sampler ΓC for $\mathcal{C} = \textsc{Seq}(\mathcal{A})$:

Generate k according to a geometric law of parameter $A(x)$

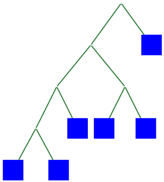
Generate a k-tuple $\langle\ \Gamma A(x)\ , \dots ,\ \Gamma A(x)\ \rangle$ $\rightarrow$ independent calls.

Remark: $A(x)$, $B(x)$ and $C(x)$ are given by an ***oracle***.

Binary trees

$$\mathcal{B} = \mathcal{Z} + \mathcal{B} \times \mathcal{B}$$

$$B(z) = z + B(z)^2 = \frac{1 - \sqrt{1-4z}}{2}$$



Algorithm: $\Gamma B(x)$

$b \leftarrow \mathrm{Bern}(x/B(x))$;
if $b = 1$ **then**
  Return ■
else
  Return $\langle\ \Gamma B(x)\ ,\ \Gamma B(x)\ \rangle$;
end if

Examples of specifications with $\{\cup, \times, \text{Seq}\}$

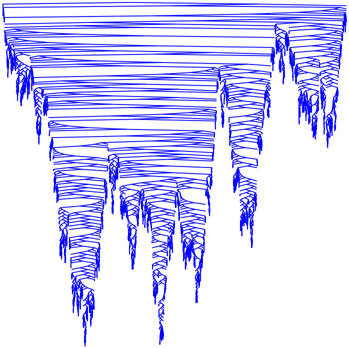
Regular specifications (non recursive).

- integer compositions, permutations,...
- polyominos that have rational g.f.: column-convex,

- regular languages,

Context-free specifications.

- any algebraic language,
- tree-like structures
  - k-ary, 2–3–4 trees, ...,
  - triangulations,
  - noncrossing graphs,
  - general planar rooted trees,
  - ...



Labelled classes

Same algorithms, with exponential generating functions

construction	sampler
$\mathcal{C} = \emptyset$ or $\mathcal{Z}$	$\Gamma C(x) :=$ ε or atom
$\mathcal{C} = \mathcal{A} + \mathcal{B}$	$\Gamma C(x) :=$ Bern $\frac{\hat{A}(x)}{\hat{C}(x)} \longrightarrow \Gamma A(x) \mid \Gamma B(x)$
$\mathcal{C} = \mathcal{A} \times \mathcal{B}$	$\Gamma C(x) := \langle\ \Gamma A(x)\ ;\ \Gamma B(x)\ \rangle$
$\mathcal{C} = \textsc{Seq}(\mathcal{A})$	$\Gamma C(x) :=$ Geom $\hat{A}(x) \Longrightarrow \Gamma A(x)$

Put the labels at the end !

1. Introduction
2. **Boltzmann model and free samplers**
   - Basic constructions
   - Labelled sets ans cycles
   - Back to unlabelled
3. Effective samplers

Labelled sets and cycles

Sets

Boltzmann sampler ΓC for $\mathcal{C} = \textsc{PSet}(\mathcal{A})$:
Generate k according to a Poisson law of parameter $A(x)$
Generate a k-tuple $\langle\ \Gamma A(x)\ ,\ldots,\ \Gamma A(x)\ \rangle$
Poisson law: $\mathbb{P}(X = k) = e^{-\lambda}\dfrac{\lambda^k}{k!}$

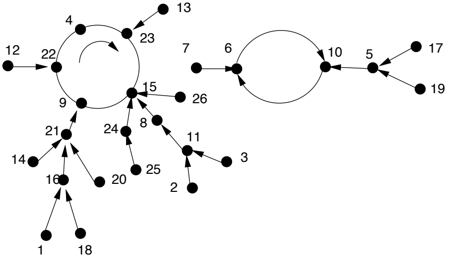
Cycles

Boltzmann sampler ΓC for $\mathcal{C} = \textsc{Cyc}(\mathcal{A})$:
Generate k according to a logarithmic law of parameter $A(x)$
Generate a k-tuple $\langle\ \Gamma A(x)\ ,\ldots,\ \Gamma A(x)\ \rangle$
Logarithmic law: $\mathbb{P}(X = k) = \dfrac{1}{\log(1-\lambda)^{-1}}\dfrac{\lambda^k}{k}$

Remark: the laws are given by simple sequential algorithms

Examples of possible labelled classes

- permutations, derangements, involutions,
- surjections,
- set partitions,
- necklaces,
- labelled (planar) trees,
- functional graphs,
- ...



1. Introduction
2. **Boltzmann model and free samplers**
   - Basic constructions
   - Labelled sets ans cycles
   - Back to unlabelled
3. Effective samplers

To begin: MSet_2 (repetitions allowed)

$\textsc{MSet}_2(\mathcal{A}) \cong$ unordered set of two objects of $\mathcal{A}$

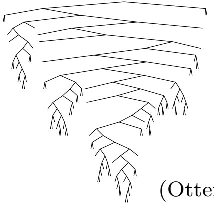
$$\mathcal{C} = \textsc{MSet}_2(\mathcal{A})$$
$$C(z) = \frac{1}{2}A^2(z) + \frac{1}{2}A(z^2) \quad \rightsquigarrow \quad \frac{1}{k}A(z^k)$$

Algorithm: $\Gamma C(x)$

if $Bern\left(\frac{1}{2}\frac{A^2(x)}{C(x)}\right) = 1$ **then**
  Return $\langle\ \Gamma A(x), \Gamma A(x)\ \rangle$
else
  $a \leftarrow \Gamma A(x^2)$;
  Return $\langle a, a \rangle$;
end if

Unlabelled binary trees

$$\mathcal{B} = \mathcal{Z} + \textsc{MSet}_2(\mathcal{B})$$

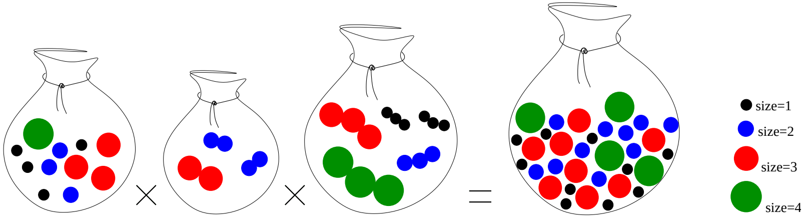


(Otter tree)

MSet: the general case (repetitions allowed)

$$\mathcal{M} = \text{MSet}(\mathcal{A}) \cong \prod_{\gamma \in \mathcal{A}} \text{Seq}(\gamma) \quad \Rightarrow \quad M(z) = \prod_{\gamma \in \mathcal{A}} (1 - z^{|\gamma|})^{-1}$$

$$M(z) = \exp\left(\sum_{k=1}^{\infty} \frac{1}{k} A(z^k)\right) = \prod_{k=1}^{\infty} \exp\left(\frac{1}{k} A(z^k)\right)$$



MSet (repetitions allowed)

Algorithm $\Gamma MSet[\mathcal{A}](x)$

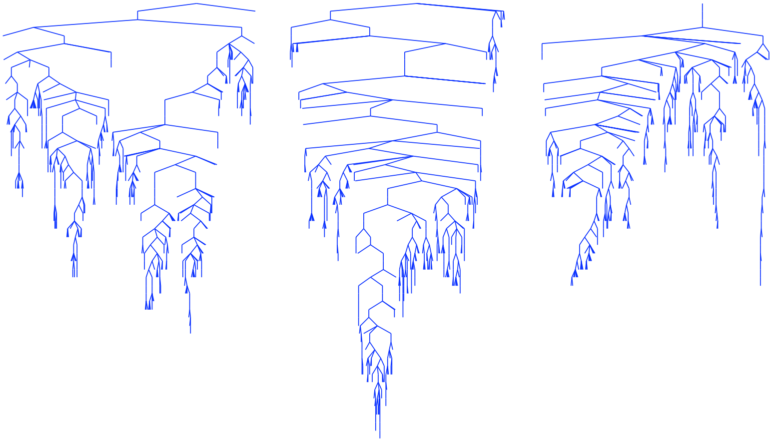
- Draw k, the max. index of a subset, depending on x;
- For each index i of a subset until $k-1$
  - Draw the number p of elements to sample, according to a Poisson law of parameter $\frac{1}{i}A(x^i)$.
  - Call $\Gamma A(x^i)$ p times, and each time, add i copies of the result to the multiset.
- for index k, draw the number p of elements to generate, according to a non zero Poisson law.

index k is drawn according to the probability distribution:

$$\Pr(K \leq k) = \prod_{j \leq k} \exp\left(\frac{1}{j} A(x^j)\right)$$

Cayley trees

$$\mathcal{T} = \mathcal{Z} \times \text{MSet}(\mathcal{T})$$



From MSet to PSet (no repetitions)

Principle : Use the following non ambiguous decomposition:

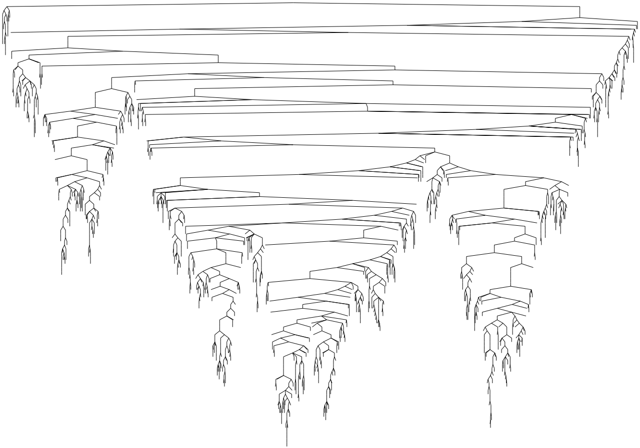
$$\textsc{MSet}(\mathcal{A}) \quad = \quad \textsc{PSet}(\mathcal{A}) \times \textsc{MSet}(\mathcal{A}^{(2)})$$

The algo. $\Gamma PSet[\mathcal{A}](x)$ to sample a powerset of objects of $\mathcal{A}$ is:

- Sample a multiset with $\Gamma MSet[\mathcal{A}](x)$,
- Extract the corresponding powerset :
  - by removing objects with even multiplicity,
  - and keeping only one occurrence of objects with **odd multiplicity**.

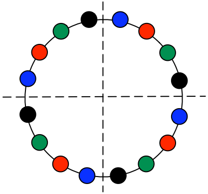
Trees without twins

$$\mathcal{T} = \mathcal{Z} \times \textsc{PSet}(\mathcal{T})$$



Cycles

$\mathcal{C} = \textsc{Cyc}(\mathcal{A}) \quad \Rightarrow \quad C(z) = \sum_{k \geq 1} \frac{\varphi(k)}{k} \log \frac{1}{1 - A(z^k)}$



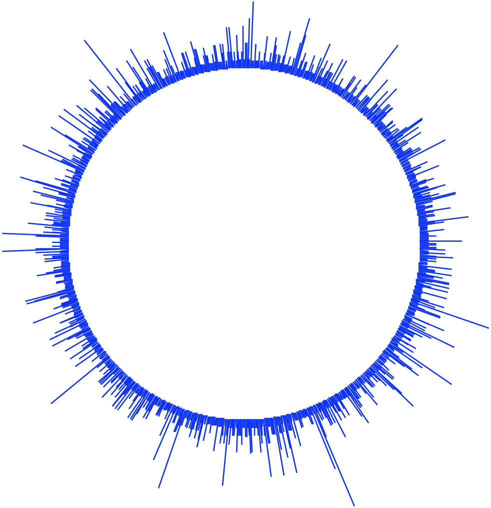
$\Gamma Cyc[\mathcal{A}](x)$

- Draw the replication order k of the cycle.
- Draw the length j of the pattern according to a logarithmic law of parameter $A(x^k)$.
- Draw the pattern m, calling $\Gamma A(x^k)$ j times.
- Return a cycle composed of k copies of m.

Cyclic compositions

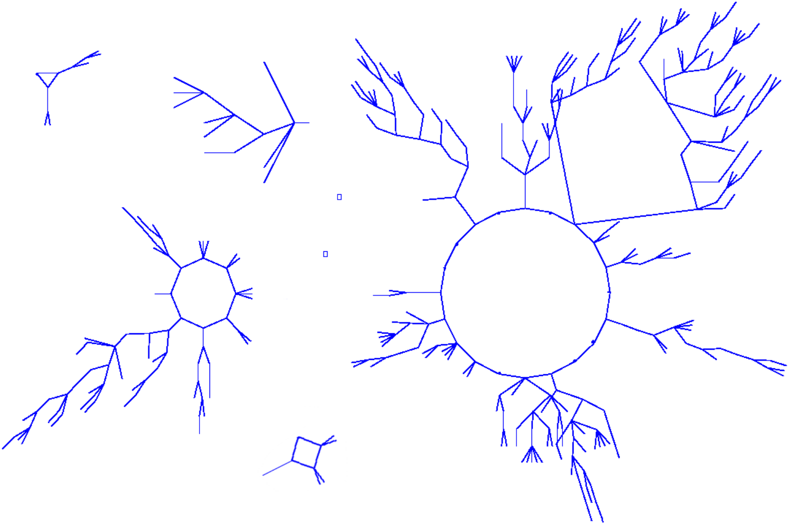
$$\mathcal{C} = \textsc{Cyc}(\mathcal{Z} \times \textsc{Seq}(\mathcal{Z}))$$

$$C(z) = \sum_{k=1}^{\infty} \frac{\varphi(k)}{k} \log \frac{1}{1 - \frac{z^k}{1-z^k}}$$



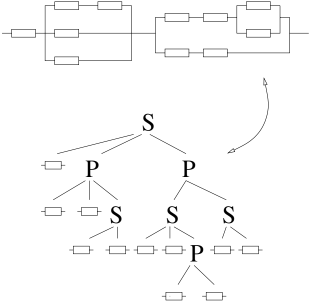
Mappings (functional graphs)

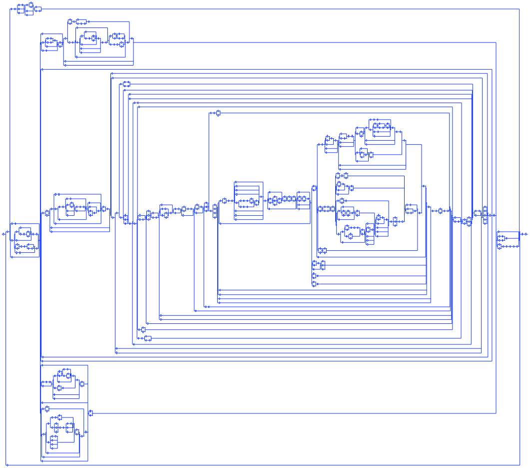
$\mathcal{G} = \textsc{Set}(\mathcal{C}), \mathcal{C} = \textsc{Cyc}(\mathcal{T}), \mathcal{T} = \mathcal{Z} \times \textsc{MSet}(\mathcal{T})$



Series-parallel circuits (cardinality constraints)

$$\begin{aligned} \mathcal{C} &= \mathcal{P} + \mathcal{S} + \mathcal{Z} \\ \mathcal{S} &= \textsc{Seq}_{\geq 2}(\mathcal{P} + \mathcal{Z}) \\ \mathcal{P} &= \textsc{MSet}_{\geq 2}(\mathcal{S} + \mathcal{Z}) \end{aligned}$$





Theorem (Free Boltzmann samplers [DuFlLo04,FlFuPi07])

For any class $\mathcal{C}$ specified (poss. recursively) using the following labelled/unlabelled constructions:

$$\varepsilon,\ \mathcal{Z},\ +,\ \times,\ \textsc{Seq},\ \textsc{Seq}_k,\ \textsc{MSet},\ \textsc{MSet}_k,\ \textsc{Cyc},\ \textsc{Cyc}_k,$$

and the labelled $\textsc{PSet}$, *the free Boltzmann sampler* $\Gamma C(x)$ *operates in linear time in the size of the object produced.*

$\textsc{PSet}$: not so bad!

if $\rho < 1$ then the *overhead* (total size of the discarded elements) is bounded by a constant.

- oracle complexity is not involved,
- size is not controlled (yet).

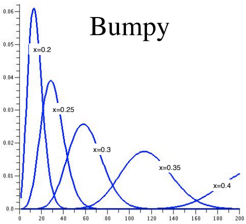
Effective samplers

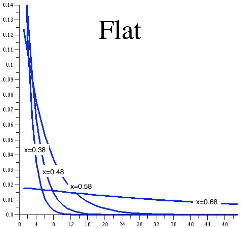
Size control – parameter tuning

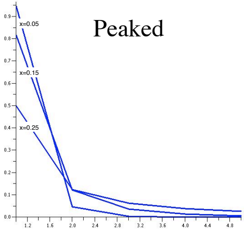
- Free samplers: produce objects with randomly varying sizes!
- Approximate and exact size samplers: use rejection.
- Tuned samplers: choose x so that expected size is n.

$$\mathbb{E}_x(N) = x\frac{C'(x)}{C(x)} \qquad \text{or} \qquad x\frac{\hat{C}'(x)}{\hat{C}(x)}$$

- Size distribution determines the cost of rejection.







Oracle [Pivoteau, Salvy, Soria 2008]

Numerical Newton iteration (step by step computation).

Binary plane trees: $B(x) = x + xY^2(x)$, e.g. $x = 0.48$,

$$Y_{k+1} = Y_k + \frac{1}{1-0.96Y_k}(0.48 + 0.48Y_k^2 - Y_k)$$

$$\begin{aligned}
Y_0 &= \mathbf{0}\\
Y_1 &= \mathbf{0}.48\\
Y_2 &= \mathbf{0}.68510385756676557863501483679525\ldots\\
Y_3 &= \mathbf{0.74}409429531735785069315411659589\ldots\\
Y_4 &= \mathbf{0.7499}4139686483588184679391778624\ldots\\
Y_5 &= \mathbf{0.7499999}411376420459420080511077\ldots\\
Y_4 &= \mathbf{0.7499999999999999}4060382090306852\ldots\\
Y_5 &= \mathbf{0.749999999999999999999999999999999}7\ldots
\end{aligned}$$

asymptotically quadratic convergence.

Proof based on Newton iteration on combinatorial structures.

Conclusion

Existing applications and related work

BaNi06 *Accessible and deterministic automata: enumeration and Boltzmann samplers*, by F. Bassino C. Nicaud. In *Fourth Colloquium on Mathematics and Computer Science*.

BoFuPi06 *Random sampling of plane partitions*, by O. Bodini, E. Fusy, and C. Pivoteau. In *GASCOM*-2006.

BoJa08 *Boltzmann samplers for colored combinatorial objects*, by O. Bodini and A. Jacquot. In *GASCOM*-2008.

DaSo07 *Degree distribution of random Apollonian network structures and Boltzmann sampling*, by A. Darrasse and M. Soria. In *International Conference on Analysis of Algorithms*, 2007, *DIMACS*.

Fusy05 *Quadratic exact-size and linear approximate-size random sampling of planar graphs*, by E. Fusy. In *International Conference on Analysis of Algorithms*, 2005, *DMTCS* Conference Volume AD (2005), pp. 125-138.

PaWe07 *Properties of Random Graphs via Boltzmann Samplers*, by K. Panagiotou and A. Weißl. In *International Conference on Analysis of Algorithms*, 2007, *DIMACS*.

Ponty06 *Modélisation de séquences génomiques structurées, génération aléatoire et application*, by Yann Ponty, PhD Thesis, Université Paris-Sud, 2006.

...

Coming soon...?

- other constructions: box operator, shuffle, ...
- multivariate Boltzmann samplers,
- oracle and automatic singularities,
- discrete samplers,
- specialized samplers,
- new applications,
- ...