Uniform Random Generation for the Powerset Construction

Paul Zimmermann
Inria Lorraine

December 12, 1994

[summary by Eithne Murray]

Abstract

An algorithm for the uniform random generation of the powerset construction is presented. Given a combinatorial class I, together with a counting procedure and an unranking procedure (or simply a random generation procedure) for I, this algorithm provides counting and unranking (or random generation) procedures for $P = \texttt{powerset}(I)$. For most combinatorial structures, each random powerset of size n is produced in $\mathcal{O}(n \log n)$ arithmetic operations in the worst case, after $\mathcal{O}(n^2)$ coefficients have been computed. This work is an extension of the algorithms developed in [1, 2], that have been implemented in the Gaïa (now combstruct) Maple package [4].

1. Introduction

Given a combinatorial class I of unlabelled objects such that for each integer n the number $I[n]$ of objects of size n is finite (and, for convenience, $I[0] = 0$), the problem is to generate uniformly at random an object of size n from $\texttt{powerset}(I)$, where $\texttt{powerset}(I)$ means the class of sets without repetition made from objects in I.

Associate with each combinatorial class I two procedures — a counting procedure `countI`, such that $\texttt{countI}(n)$ gives the number $I(n)$ of objects of size n, and an unranking procedure `unrankI` which implements a bijection between $[0, I(n) - 1]$ and the set of objects of size n from I. Thus $\texttt{unrankI}(n, k)$ returns the object of size n whose *rank* is k.

Given these two procedures, the algorithm constructs similar functions `countP` and `unrankP` to count and generate at random the objects from $P = \texttt{powerset}(I)$.

The article [5] this seminar is based on is available from http://www.loria.fr/~zimmerma/gaia. It contains implementation details, proofs, examples and experimental results.

2. The Counting Problem

The generating functions $P(z)$ and $I(z)$ satisfy this identity due to Pólya: $P(z) = \exp(I(z) - \frac{1}{2}I(z^2) + \frac{1}{3}I(z^3) - \frac{1}{4}I(z^4) + \cdots)$. Using this identity and the operator $\Theta = z\, d/dz$ (see [1]), it is easy to obtain equations to compute the coefficients $P[k] = [z^k]P(z)$ for $k = 1, ..., n$ in $\mathcal{O}(n^2)$ operations.

3. An Unranking Algorithm

To generate a random object of size n from a powerset, consider the problem in two steps. First, generate the *shape* the generated object will have, that is, how many objects of each size will be present in the set. Second, for each size of object in the set, generate the objects of that size.

3.1. Shape Generation. To generate a powerset of size n, the algorithm chooses non-negative integers $(i_1, \ldots, i_k)$ such that $n = i_1 + 2i_2 + \cdots + ki_k$. It must ensure that the shape is generated with a probability based on the number of powersets having the shape $(i_1, \ldots, i_k)$. This probability depends only on the numbers $I[\ell]$, for ℓ between 1 and n. Thus, only the procedure `countI` is needed to solve the shape generation problem. This algorithm is based on the decomposition $P_{\ell,m} = \texttt{Prod}(P_{\ell,2m}, P_{\ell+m,2m})$, where $P_{\ell,m}$ is the set of objects of size $\ell + \lambda m$, where λ is a non-negative integer, m is a power of 2, $1 \le \ell \le m$, and `Prod` denotes the cartesian product. It involves calculating $\mathcal{O}(n^2)$ sizes of $P_{\ell,m}$.

3.2. Equal Size Generation. The equal size generation problem is equivalent to the problem of selecting k distinct elements $a_1, \ldots, a_k$ from $1, 2, \ldots, n$ (called "selection sampling" by Knuth [3]). There appears to be no known algorithm to do unranking for this problem in $\mathcal{O}(k)$ time and space. P. Zimmermann proposes an unranking algorithm for the selection sampling problem that has $\mathcal{O}(k \log n)$ worst case complexity.

A method of unranking powersets is then obtained by using this selection sampling algorithm. By replacing the unranking procedure for I in the unranking algorithm by a random generation procedure, a random generation procedure for P is also formed. The complexity analysis for the unranking algorithm depends on a condition of *standard growth* on the combinatorial class, while the analysis for the random generation algorithm holds for all combinatorial classes. Most combinatorial classes satisfy this condition, but otherwise, such as for a recursive structure where I depends on P, there is little information about the algorithm's efficiency.

DEFINITION 1. A combinatorial class I is of *standard growth* if their exists a constant A such that the number $I[n]$ of structures of size n satisfies $I[n] \le n^{An}$ for n sufficiently large.

THEOREM 1. *If I is any combinatorial class, and* `randomI`(n) *has average cost $\mathcal{O}(n \log n)$, then* `randomP`(n, k) *also has average cost $\mathcal{O}(n \log n)$. If I is a combinatorial class of standard growth and* `unrankI`(n, k) *has worst case $\mathcal{O}(n \log n)$, then* `unrankP`(n, k) *also has worst case cost $\mathcal{O}(n \log n)$.*

4. Conclusion and Open Questions

Given any combinatorial class I, a counting procedure for I, and a procedure for unranking (random generation) for I, there is an algorithm to do unranking and random generation for $P = \texttt{powerset}(I)$. Some questions remain. Is there an $\mathcal{O}(k)$ algorithm for unranking k-samples in an n-set? And what is the worst case complexity of this algorithm when I has a recursive specification? Also, the pre-processing time of this algorithm is rather high ($\mathcal{O}(n^2)$ operations and about $\mathcal{O}(n^4)$ time). This pre-processing cost should be reduced.

Bibliography

[1] Flajolet (Philippe), Zimmerman (Paul), and Van Cutsem (Bernard). – A calculus for the random generation of labelled combinatorial structures. *Theoretical Computer Science*, vol. 132, n° 1-2, 1994, pp. 1–35.

[2] Flajolet (Philippe), Zimmermann (Paul), and Van Cutsem (Bernard). – A calculus of random generation: Unlabelled structures. In preparation.

[3] Knuth (Donald E.). – *The Art of Computer Programming*. – Addison-Wesley, 1981, 2nd edition, vol. 2: Seminumerical Algorithms.

[4] Zimmermann (Paul). – Gaïa: A package for the random generation of combinatorial structures. *MapleTech*, vol. 1, n° 1, 1994, pp. 38–46.

[5] Zimmermann (Paul). – Uniform random generation for the powerset construction. In Leclerc (B.) and Thibon (J. Y.) (editors), *Formal power series and algebraic combinatorics*. pp. 589–600. – Université de Marne-la-Vallée, 1995. Proceedings SFCA'95.