# Lexicographic unranking algorithms for the Twelvefold Way

*Amaury Curiel* & Antoine Genitrini
SU - LIP6

Wednesday, June 19th

# Table des matières

# The Twelvefold Way

# The Twelvefold Way

| elts of $\mathcal{N}$ | elts of $\mathcal{K}$ | $f$ is arbitrary | $f$ is injective | $f$ is surjective |
|---|---|---|---|---|
| dist. | dist. | **1.**<br>$n$-sequence in $\mathcal{K}$ | **2.**<br>$n$-permutation of $\mathcal{K}$ | **3.**<br>composition of $\mathcal{N}$<br>with $k$ subsets |
| enumeration | | $k^n$ | $k^{\underline{n}}$ | $k! \cdot \left\{ {n \atop k} \right\}$ |
| indist. | dist. | **4.**<br>$n$-multisubset of $\mathcal{K}$ | **5.**<br>$n$-subset of $\mathcal{K}$ | **6.**<br>composition of $n$<br>with $k$ terms |
| enumeration | | $\binom{k+n-1}{n}$ | $\binom{k}{n}$ | $\binom{n-1}{n-k}$ |
| dist. | indist. | **7.**<br>partition of $\mathcal{N}$<br>into $\leq k$ subsets | **8.**<br>partition of $\mathcal{N}$<br>into $\leq k$ elements | **9.**<br>partition of $\mathcal{N}$<br>into $k$ subsets |
| enumeration | | $\sum_{i=0}^{k} \left\{ {n \atop i} \right\}$ | $[n \leq k]$ | $\left\{ {n \atop k} \right\}$ |
| indist. | dist. | **10.**<br>partition of $n$<br>into $\leq k$ parts | **11.**<br>partition of $n$<br>into $\leq k$ parts $\{1\}$ | **12.**<br>partition of $n$<br>into $k$ parts |
| enumeration | | $p_k(n+k)$ | $[n \leq k]$ | $p_k(n)$ |

Figure 1: Classification due to Rota in the 60's

# The Twelvefold Way (2)

**About generation**

The counting enumeration formula is well known for each of the twelve cases.
What about the generation algorithm for these combinatorial objects?

# Unranking algorithms

**Definition**

Unranking algorithms are used to generate combinatorial objects.

- Let $\mathcal{C}$ be a class of combinatorial objects
  - Let $\mathcal{O}$ be a total order over $\mathcal{C}$
  - Let $r \in \mathbb{N}$ such that $r < |\mathcal{C}|$
- An unranking algorithm for $\mathcal{C}$ outputs the $r^{\text{th}}$ object of $\mathcal{C}$ according to $\mathcal{O}$

**Uses**

Bijection between $[\![0, |\mathcal{C}| - 1]\!] = [\![|\mathcal{C} - 1|]\!]$ and $\mathcal{C}$

2 main use cases :

- Exhaustive generation
- Random generation

**Unranking algorithms for the Twelvefold Way**

Given a constructive enumeration formula $f$ for a class of combinatorial objects, we derive an unranking algorithm.

# Unranking algorithms for the Twelvefold Way: Example

**Unranking algorithms for the Twelvefold Way: Example**

We are interested in combinations of $k$ elements among $[\![0, n-1]\!]$.

- Such objects are counted by the binomial coefficient $\binom{n}{k}$
  - $\binom{n}{k} = \begin{cases} \binom{n}{0}=\binom{n}{n}=1 \\ \binom{n-1}{k-1}+\binom{n-1}{k} \end{cases}$
  - Combinatorially, this translates to :
    - The first element of the set is in the combination, in this case it remains $k-1$ elements to choose among $n-1$ elements
    - The first element is not in the combination, in this case it remains $k$ elements to choose among $n-1$ elements.
  - The unranking algorithm is then straightforward :
    - If $r < \binom{n-1}{k-1}$, the first element is in the combination, otherwise it is not.
    - After having adapted the rank, we can recursively build the rest of the combination.

# Unranking algorithms for the Twelvefold Way

| Rank | Combinations |
|---|---|
| 0 | (1 2 3) |
| 1 | (1 2 4) |
| 2 | (1 2 5) |
| 3 | (1 3 4) |
| 4 | (1 3 5) |
| ... | ... |

# Unranking algorithms for the Twelvefold Way

| Rank | Combinations | Set partitions |
|---|---|---|
| 0 | (1 2 3) | { { 1 4 5 } { 2 } { 3 } } |
| 1 | (1 2 4) | { { 1 5 } { 2 4 } { 3 } } |
| 2 | (1 2 5) | { { 1 5 } { 2 } { 3 4 } } |
| 3 | (1 3 4) | { { 1 3 5 } { 2 } { 4 } } |
| 4 | (1 3 5) | { { 1 5 } { 2 3 } { 4 } } |
| ... | ... | ... |

**Order**

Can we fix an order that is more intuitive?

# Unranking algorithms for the Twelvefold Way (3)

## Lexicographic order

- There exist a lexicographic unranking algorithm for nine out of the twelve cases of the Twelvefold Way

| elts of $\mathcal{N}$ | elts of $\mathcal{K}$ | $f$ is arbitrary | $f$ is injective | $f$ is surjective |
|---|---|---|---|---|
| dist. | dist. | **1.** n-sequence in $\mathcal{K}$ | **2.** n-permutation of $\mathcal{K}$ | **3.** composition of $\mathcal{N}$ with $k$ subsets |
| enumeration | | $k^n$ | $k^{\underline{n}}$ | $k! \cdot \left\{ {n \atop k} \right\}$ |
| indist. | dist. | **4.** $n$-multisubset of $\mathcal{K}$ | **5.** $n$-subset of $\mathcal{K}$ | **6.** composition of $n$ with $k$ terms |
| enumeration | | $\binom{k+n-1}{n}$ | $\binom{k}{n}$ | $\binom{n-1}{n-k}$ |
| dist. | indist. | **7.** partition of $\mathcal{N}$ into $\leq k$ subsets | **8.** partition of $\mathcal{N}$ into $\leq k$ elements | **9.** partition of $\mathcal{N}$ into $k$ subsets |
| enumeration | | $\sum_{i=0}^{k} \left\{ {n \atop i} \right\}$ | $[n \leq k]$ | $\left\{ {n \atop k} \right\}$ |
| indist. | indist. | **10.** partition of $n$ into $\leq k$ parts | **11.** partition of $n$ into $\leq k$ parts $\{1\}$ | **12.** partition of $n$ into $k$ parts |
| enumeration | | $p_k(n+k)$ | $[n \leq k]$ | $p_k(n)$ |

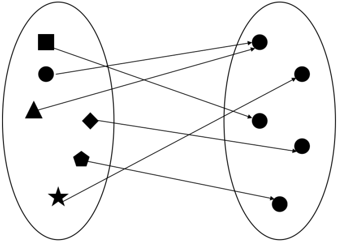
# Lexicographic unranking of set partitions [case 9]

# Set partitions

**Set partitions**

Giving a set of $n$ labeled elements, and $k$ indistinguishable blocks, a set partition is a way to distribute the elements among the blocks such that:

- Each element is exactly in one block
- There is no empty block



{ {■}{● ▲}{◆}{⬟}{★} }

# Set partitions (2)

**Prop. 1:** Set partitions enumeration

The set partitions [9] are counted by the Stirling numbers of the second kind $\left\{ {n \atop k} \right\}$ recursively defined by : $\left\{ {n \atop k} \right\} = \begin{cases} 1 & \text{if } k=1 \vee n=k \\ \left\{ {n-1 \atop k-1} \right\} + k \cdot \left\{ {n-1 \atop k} \right\} & \text{otherwise} \end{cases}$

**Sequential form [Mansour, 2012]**

In the following, we will represent a set partition as a sequence of blocks ordered with respect to their minimal element.

- $1/23/45$ is the sequential form of $\{\{2,3\}\{1\}\{4,5\}\}$

**Theorem 1:** Lexicographic order

Let $P$ and $Q$ be two subsets of integers. We say that $P \leq Q$ iff either

$$\begin{cases} P=Q \quad \vee \quad P \subset Q \text{ and } \max(P) < \min(Q/P) \\ Q \subset P \text{ and } \min(P/Q) < \max(Q) \quad \vee \quad \min(P/Q) < \min(Q/P) \end{cases}$$

This relation is a total order over subsets of integers. Futhermore, considering sets partitions in sequential form, the derived total order is the **lexicographic order**.

# Set partitions (3)

| Rank | Partition | Canonical form [Mansour, 2012] | Rank | Partition | Canonical form [Mansour, 2012] |
|---|---|---|---|---|---|
| 0 | 1/2/345 | 12333 | 13 | 13/2/45 | 12133 |
| 1 | 1/23/45 | 12233 | 14 | 13/24/5 | 12123 |
| 2 | 1/234/5 | 12223 | 15 | 13/25/4 | 12132 |
| 3 | 1/235/4 | 12232 | 16 | 134/2/5 | 12113 |
| 4 | 1/24/35 | 12323 | 17 | 135/2/4 | 12131 |
| 5 | 1/245/3 | 12322 | **18** | **14/2/35** | **12313** |
| 6 | 1/25/34 | 12332 | **19** | **14/23/5** | **12213** |
| 7 | 12/3/45 | 11233 | 20 | 14/25/3 | 12312 |
| 8 | 12/34/5 | 11223 | 21 | 145/2/3 | 12311 |
| **9** | **12/35/4** | **11232** | 22 | 15/2/34 | 12331 |
| **10** | **123/4/5** | **11123** | 23 | 15/23/4 | 12231 |
| 11 | 124/3/5 | 11213 | 24 | 15/24/3 | 12321 |
| 12 | 125/3/4 | 11231 | | | |

Table 3: Ranking of the 3-partitions of ⟦5⟧

# Set partitions (4)

**Unranking in lexicographic order**

The combinatorial interpretation of the construction of Stirling numbers of the second kind does not lead to lexicographic order.

- How can we lexicographically unrank a set partition?

# Prefix of a set partition

**Prefix of a set partition**

Let $P = p_1/\ldots/p_k$ be a set partition in sequential form. We say that $\pi$ is a prefix of $P$ iff $\pi \subset p_1$ and $\pi \leq p_1$

**Example**

1, 12 and 123 are prefixes of 123/4/56

# Prefix completion theorem

**Theorem 2:** Prefix completion

Let $1 \leq k \leq n$ be two positive integers, Let $l$ and $d$ be two integers such that either $l = d = 1$ or $1 < l \leq d$. For a given prefix $\Pi = \alpha_1, ..., \alpha_{l-1}, d$, we define $S_k^n(l, d)$ to be the numbers of partitions in $\mathcal{P}_k^n$ accepting $\Pi$ as prefix of length $l$. Futhermore, we have :
$S_k^n(l, d) = \sum_{u=0}^{n-k-l+1} \left\{ {n-l-u \atop k-1} \right\} \cdot \binom{n-d}{u}$

# Prefix completion theorem : Interprétation

## Combinatorial interpretation

$S_k^n(l,d) = \sum_{u=0}^{n-k-l+1} \left\{ {n-l-u \atop k-1} \right\} \cdot \binom{n-d}{u}$:

Let $P$ be a set partition of $\mathcal{P}_k^n$ admiting $\Pi$ as prefix.

- $u$ is the numbers of elements that we add to the prefix to complete it.
  - If $\Pi$ is the first block of $P$, then, $u = 0$.
  - If $\Pi$ is a strict prefix of $P$, then the maximal capacity of the first block of $P$ is $n - k$. Therefore, we can't add more than $n - k - l + 1$ elements to the prefix.
- Once the size of the prefix is computed, we need to determine the number of ways to complete the prefix.
  - There are $u$ elements to choose among the legals elements to complete the prefix. This is counted by $\binom{n-d}{u}$.
- Once the prefix is completed, we need to determine the remainings blocks of $P$.
  - This is counted by $\left\{ {n-l-u \atop k-1} \right\}$

# Prefix completion theorem : Other forms

**Other forms**

The prefix completion theorem can be rewritten in other forms :

- $\tilde{S}_k^n(d) = \sum_{u=0}^{\min(n-k,n-d)} \left\{ {n-u \atop k} \right\} \cdot \binom{n-d}{u} = S_{k+1}^{n+l}(d+l)$
- $\tilde{S}_k^n(d) = \sum_{u=0}^{\min(n-k,d)} (-1)^u \left\{ {n+1-u \atop k+1} \right\} \binom{d}{u}$

# Algorithm

# Algorithm design

## Algorithm design

- We build the set partition block by block from left to right
- We build the blocks from the left to right
- In order to build the $i^{\text{th}}$ block, we use the $\tilde{S}_k^n$ formula to determine the prefix starting by the prefix 1 and ending by the prefix $1n$.

## Example

We want to unrank the $1^{\text{st}}$ block of the $20^{\text{th}}$ partition of $[\![5]\!]$ in 3 blocks.

| Step | Rank | Block | Prefix tested | Number of partitions | Comment |
|---|---|---|---|---|---|
| 1 | 20 | {} | 1 | 25 | $20 < 25$ |
| 1 bis | 20 | {1} | exactly 1 | 6 | $20 > 6$ |
| 2 | 14 | {1} | 12 | 6 | $14 > 6$ |
| 3 | 8 | {1} | 13 | 5 | $8 > 5$ |
| 4 | 3 | {1} | 14 | 4 | $3 < 4$ |
| 5 | 3 | {1,4} | exactly 14 | 3 | $3 \leq 3$ |

# Algorithm design (2)

**Optimisation**

Can we do better ?

**Optimisation**

YES !

- We can use the $R_k^n$ formula in order to use a binary search to determine the prefix.

**Theorem 3:** An other form of the prefix completion theorem

Let $1 \leq k \leq n$ be two positive integer. Let $d_1 \in [\![2, n]\!]$, $d_0 \in [\![d_1 - 1]\!]$ and $l > 1$ be integers. For a given prefix $\alpha_1, ..., \alpha_{l-2}, d_0$, the numbers of elements of $\mathcal{P}_k^n$ that admit a length-$l$ prefix satisfying $1, \alpha_2, ..., \alpha_{l-2}, d_0, \widetilde{d_1}$, $\widetilde{d_1} \in [\![d_0 + 1, d_1]\!]$ is given by :

$R_k^n(l, d_0, d_1) = \tilde{S}_{k-1}^{n-l}(d_0 - l) - \tilde{S}_{k-1}^{n-l}(d_1 + 1 - l)$

# Algorithm design (3)

**Example**

We want to unrank the $1^{st}$ block of the $20^{th}$ partition of $[\![5]\!]$ in 3 blocks.

| Step | Rank | Block | Prefix tested | Number of partitions |
|---|---|---|---|---|
| 1 | 20 | {} | 13 | 17 |
| 2 | 3 | {} | 14 | 4 |
| 3 | 3 | {1,4} | - | - |

Our algorithm converges in 3 steps instead of 5.

# Algorithm implementation

**Question**

Can we do better ?

**Optimisation**

YES !

- The $R_k^n$ formula has several writing in the form of sums wich has different numbers of terms. Thus, before calling $R_k^n$, we choose the formula that has the least number of terms.

# Algorithm implementation

## Algorithm implementation

In order to compute the Stirling numbers, we had two main choices:

- Put in cache the values of the Stirling numbers
- Compute them on the fly

## Caching Stirling numbers

When $k \approx \log(n)$, $\log\left(\left\{ n \atop k \right\}\right) = \Theta\left(n \cdot \log(n)^5\right)$ [Rennie & Dobson, 1969]. Thus, precomputing the Stirling numbers cost among $O(n^3)$ bit memory space, which will saturate the memory of a modern laptop for $n \approx 3000$.

## Computing Stirling numbers on the fly

The explicit formula of the Stirling numbers of the second kind is $\left\{ n \atop k \right\} = \frac{1}{k!} \sum_{i=0}^{k} (-1)^{k-i} \binom{k}{i} i^n$.

which costs $O(k)$ operations on big int. to compute $\left\{ n \atop k \right\}$. In practice, the cost of such computation is too expensive for $n \approx 2000$.

# Algorithm implementation (2)

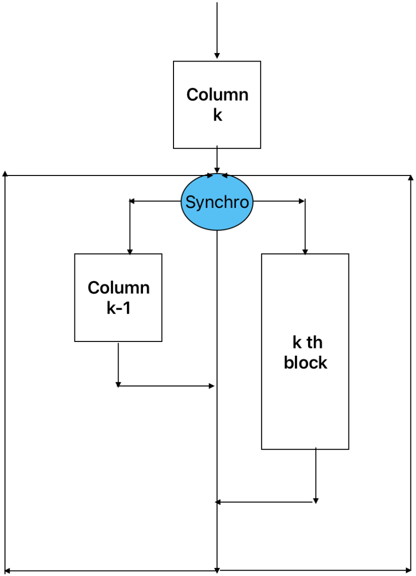
**Usefull Stirling numbers**

During the $k^{\text{th}}$ block construction: only need the $k^{\text{th}}$ and the $(k-1)^{\text{th}}$ columns. Futhermore, compute the previous Stirling column from the current one is linear using the recursive formula.

**Number of call to $R_k^n$**

Determining one component of the $k^{\text{th}}$ block costs $O(\log(n))$ calls to $R_k^n$. Futhermore, the cost of a call to $R_k^n$ is $O(n)$.

# Algorithm implementation



# Complexity analysis & performance in practice

**Theorem 4:** Theoretical complexity

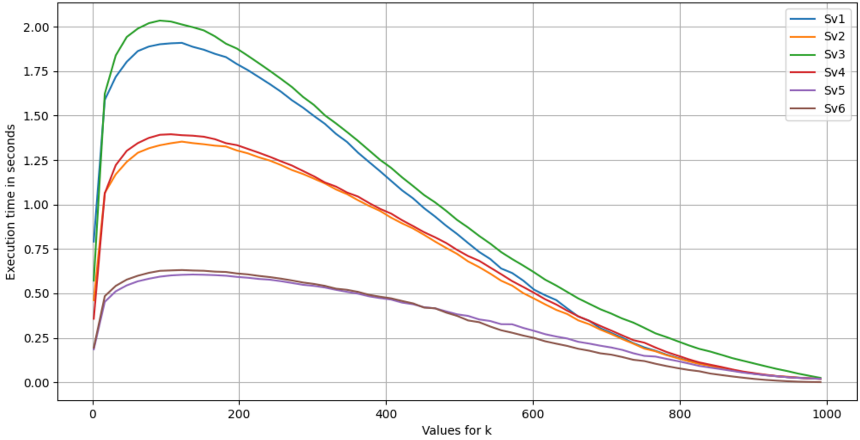
The bit-complexity of our algorithm is bounded by
$$O\left(\frac{(n-k)^3 \cdot M(n)}{n} \ln(n) \ln(k) + \frac{k(n-k)^2 M(n)}{n} \ln(n) \left(\ln\left(\frac{n \cdot e}{k}\right)\right)\right)$$
where $M(n)$ is the cost of a multiplication of two $n$-bits big integers (in practice $M(n) = O\left(n^{\log(3)}\right)$).

**Benchmark procedure**

In order to have an overview of the performance of our algorithm, we provide 6 diferents implementations of $R_k^n$ :

- S_v1: The direct implementation of $R_k^n$.
- S_v2: S_v1 but taking into acount the symmetry of binomial coefficients.
- S_v3: The binomial transform of $R_k^n$.
- S_v4: S_v3 but taking into acount the symmetry of binomial coefficients.
- S_v5: Call to S_v2 or S_v4 depending on the number of terms of the formula.
- S_v6: S_v5 but Stirling numbers are precomputed.

# Performance in practice



# Extension to problems [case 3] and [case 7]

# Ordered set partitions [3]

**Prop. 2:** Ordered set partitions enumeration

The ordered set partitions are counted by the **ordered Stirling numbers of the second kind** defined by : $\mathcal{O}_k^n = k! \cdot \left\{ {n \atop k} \right\}$.

**Theorem 5:** Prefix completion

Let $1 \leq k \leq n$ be two positive integers, let $l$ and $d$ be two integers. For a given prefix $\Pi = \alpha_1, ..., \alpha_{l-1}, d$, we define $T_{k(l,d)}^n$ to be the number of ordered set partitions of $[\![n]\!]$ in $k$ parts that admit $\Pi$ as prefix of length $l$. We have

$T_{k(l,d)}^n = \sum_{u=0}^{n-k-l+1} k! \cdot \left\{ {n-l-u \atop k-1} \right\} \cdot \binom{n-d}{u}$.

# Bell's set partitions

**Prop. 3:** Bell set partitions

The Bell set partitions are counted by the **Bell numbers** $B_n$ defined by :
$B_n = \sum_{k=1}^{n} \left\{ {n \atop k} \right\}$.

**Theorem 6:** Prefix completion

Let $1 \leq k \leq n$ be two positive integers, let $l$ and $d$ be two integers such that either $l = d = 1$ or $1 < l \leq d$. For a given prefix $\Pi = \alpha_1, \ldots, \alpha_{l-1}, d$, we define $S_k^n(l, d)$ to be the numbers of partitions in $\mathcal{P}_k^n$ accepting $\Pi$ as prefix of length $l$. Futhermore, we have:
$S_k^n(l, d) = \sum_{u=0}^{n-k-l+1} \sum_{i=1}^{k} \left\{ {n-l-u \atop i-1} \right\} \cdot \binom{n-d}{u}$.

# Conclusion

# Conclusion

## Conclusion

In this talk, we have presented a prefix completion theorem which leads to a lexicographic unranking algorithm for set partitions. This algorithm is highly adaptatable to other combinatorial structures.

- Code available at https://pkg.golang.ir/github.com/AMAURYCU/setpartition_unrank#section-readme

## Future works

- Our algorithm is easily adaptable to other structures to handle them, we need to change few lines of code.
  - How can we generalize our algorithm so that we do not have to change any lines of code?
- Try to improve the efficiency for other existing unranking algorithms using concurrency.

# Bibliographie

Mansour, T. [2012]. *Combinatorics of Set Partitions*. Taylor & Francis. https://doi.org/10.1201/b12691

Rennie, B., & Dobson, A. [1969]. On stirling numbers of the second kind. *Journal of Combinatorial Theory*, *7*[2], 116–121. https://doi.org/10.1016/S0021-9800(69)80045-1

| elts of $\mathcal{N}$ | elts of $\mathcal{K}$ | $f$ is arbitrary | $f$ is injective | $f$ is surjective |
|---|---|---|---|---|
| dist. | dist. | **1.** n-sequence in $\mathcal{K}$ | **2.** n-permutation of $\mathcal{K}$ | **3.** composition of $\mathcal{N}$ with $k$ subsets |
| enumeration | | $k^n$ | $k^{\underline{n}}$ | $k! \cdot \left\{ {n \atop k} \right\}$ |
| indist. | dist. | **4.** $n$-multisubset of $\mathcal{K}$ | **5.** $n$-subset of $\mathcal{K}$ | **6.** composition of $n$ with $k$ terms |
| enumeration | | $\binom{k+n-1}{n}$ | $\binom{k}{n}$ | $\binom{n-1}{n-k}$ |
| dist. | indist. | **7.** partition of $\mathcal{N}$ into $\leq k$ subsets | **8.** partition of $\mathcal{N}$ into $\leq k$ elements | **9.** partition of $\mathcal{N}$ into $k$ subsets |
| enumeration | | $\sum_{i=0}^{k} \left\{ {n \atop i} \right\}$ | $[n \leq k]$ | $\left\{ {n \atop k} \right\}$ |
| indist. | indist. | **10.** partition of $n$ into $\leq k$ parts | **11.** partition of $n$ into $\leq k$ parts $\{1\}$ | **12.** partition of $n$ into $k$ parts |
| enumeration | | $p_k(n+k)$ | $[n \leq k]$ | $p_k(n)$ |