Génération des classes combinatoires étiquetées

Katia A., Firat M.

Sorbonne Université, UFR Ingénierie
Frédéric PESCHANSKI

Novembre 2019

Sommaire

Introduction

Titre : A generic approach for the unranking of labeled combinatorial classes
Auteurs : Conrado Martínez Xavier Molinero

But : Explication détaillée des algorithmes d'unranking présentés

Classe combinatoire

Une classe combinatoire est un ensemble d'objets muni d'une application appelée "taille" qui associe à chacun de ses éléments un entier naturel.

Soit $\mathscr{A}$ une classe combinatoire et $\mathscr{A}_n$ un sous ensemble de $\mathscr{A}$ ne contenant que les objets de taille n. $\forall n \in \mathbb{N}, \mathscr{A}_n$ est un ensemble fini.

Exemple : chaînes de caractères et arbres binaires.

3 Problèmes

1. Génération ordonnée : itérateur générant tous les objets d'une taille donnée d'une classe combinatoire $\mathscr{A}$.
2. Ranking : donner le rang d'un objet *a* appartenant à une classe combinatoire $\mathscr{A}$.
3. **Unranking** : Générer un objet *a* selon une classe combinatoire $\mathscr{A}$ et un rang donnés.

Unranking

Génération d'un objet a d'une classe combinatoire $\mathscr{A}$ selon :

- Spécification de $\mathscr{A}$
- Un rang

Rang de a dans $\mathscr{A}$: nombre d'objets de même taille dans $\mathscr{A}$ qui sont strictement plus petits que a.

Spécification

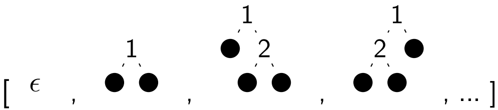
$$\mathscr{A}_i = \Psi_i(\mathscr{A}_{j_0}^{(i)}, \ldots, \mathscr{A}_{j_i}^{(i)})$$

- Ψ_i : Les opérations admissibles
- $\mathscr{A}_d^{(i)}$:
  - classe-ϵ
  - classe atomique
  - $\forall d, d' \in \{j_0, ..., j_i\}$ si $d \neq d'$ alors $\mathscr{A}_{j_d}^{(i)} \neq \mathscr{A}_{j_{d'}}^{(i)}$

Exemple de classe combinatoire

Soit $\mathscr{A}$ une classe combinatoire d'arbre binaire et $a \in \mathscr{A}$.

- Spécification : $\mathscr{A} = \epsilon + Z * \{\mathscr{A}, \mathscr{A}\}$
- taille(a) $=$ # d'atomes dans a
- rang(a) $=$ taille(a)



Spécification standard

- Union
- Produit
- Produit en boîte

Ordre : Union

Soit $\mathscr{A}$ et $\mathscr{B}$ deux classes combinatoires et $<_{\mathscr{C}_n}$ l'ordre fixé pour les objets de taille n dans $\mathscr{C}$.
Si $\mathscr{C} = \mathscr{A} \cup \mathscr{B}$ alors les éléments de $\mathscr{A}$ apparaissent avant ceux de $\mathscr{B}$ dans $\mathscr{C}$.

$$\begin{aligned}\gamma_1 <_{\mathscr{C}_n} \gamma_2 \iff\ & (\gamma_1 <_{\mathscr{A}_n} \gamma_2 \text{ et } \gamma_1, \gamma_2 \in \mathscr{A}) \text{ ou} \\ & (\gamma_1 <_{\mathscr{B}_n} \gamma_2 \text{ et } \gamma_1, \gamma_2 \in \mathscr{B}) \text{ ou} \\ & (\gamma_1 \in \mathscr{A} \text{ et } \gamma_2 \in \mathscr{B})\end{aligned}$$

Algorithme : Union

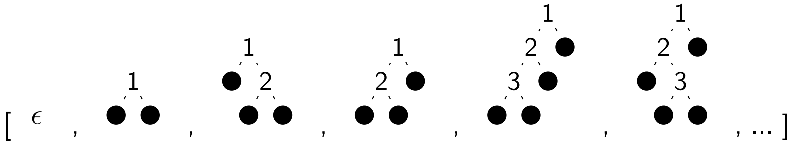
Soit $\mathscr{A} + \mathscr{B}$ l'union, **n** un rang, **i** le *i*-ème élément de rang **n** dans l'union et **count($\mathscr{A}$, n)** le nombre d'objets de rang **n** dans $\mathscr{A}$.

Algorithm 1 unrank($\mathscr{A} + \mathscr{B}$, n, i)

```
c ← count(𝒜, n)
if i < c then
    unrank(𝒜, n, i)
else
    unrank(ℬ, n, i − c)
end if
```

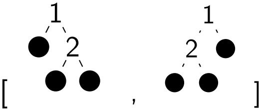
Exemple

Soit $\mathscr{A} = \mathscr{B} =$



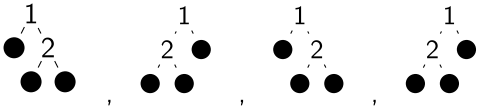
Deux structures d'arbres binaires avec étiquetage croissant.
$\mathscr{A}_n$ et $\mathscr{B}_n$ représentent les arbres binaires ayant n nœuds internes étiquetés.

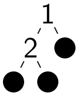
Exemple : $\mathscr{A}_2 = \mathscr{B}_2 =$

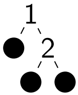


Exemple : Union

Soit $\mathscr{C} = \mathscr{A} \cup \mathscr{B}$.

$\mathscr{C}_2 = [$  $]$

$\text{unrank}(\mathscr{A} \cup \mathscr{B}, 2, 1) = \text{unrank}(\mathscr{A}, 2, 1) =$ 

$\text{unrank}(\mathscr{A} \cup \mathscr{B}, 2, 2) = \text{unrank}(\mathscr{B}, 2, 0\ (= 2 - 2)) =$ 

Ordres : Produit

Soit $\mathscr{A}$ et $\mathscr{B}$ deux classes combinatoires, $<_{\mathscr{C}_n}$ l'ordre fixé pour les objets de taille n dans $\mathscr{C}$ et $<_{\mathscr{L}_n}$ l'ordre numérique entre les étiquettes.

Soit $\mathscr{C} = \mathscr{A} \star \mathscr{B}$

- **Ordre lexicographique**
  $\mathscr{C}_n = \mathscr{A}_0 \star \mathscr{B}_n + \mathscr{A}_1 \star \mathscr{B}_{n-1} + \ldots + \mathscr{A}_n \star \mathscr{B}_0$
- **Ordre boustrophedonic**
  $\mathscr{C}_n = \mathscr{A}_0 \star \mathscr{B}_n + \mathscr{A}_n \star \mathscr{B}_0 + \mathscr{A}_1 \star \mathscr{B}_{n-1} + \mathscr{A}_{n-1} \star \mathscr{B}_1 + \mathscr{A}_2 \star \mathscr{B}_{n-2} + \ldots$

Ordre lexicographique

$\mathscr{A}$ et $\mathscr{B}$ deux classes combinatoires, $<_{\mathscr{C}_n}$ l'ordre fixé pour les objets de taille n dans $\mathscr{C}$ et $<_{\mathscr{L}_n}$ l'ordre numérique entre les étiquettes.

$$\begin{aligned}\gamma = (\alpha, \beta) <_{\mathscr{C}_n} \gamma' = (\alpha', \beta') \Longleftrightarrow\ & |\alpha| < |\alpha'| \text{ ou} \\ & (j = |\alpha| = |\alpha'| \text{ et } \alpha <_{\mathscr{A}_j} \alpha') \text{ ou} \\ & (\alpha = \alpha' \text{ et } \beta <_{\mathscr{B}_{n-j}} \beta') \text{ ou} \\ & (\alpha = \alpha' \text{ et } \beta = \beta' \text{ et } l_\gamma <_{\mathscr{L}_n} l_{\gamma'})\end{aligned}$$

Algorithme : ordre lexicographique

Algorithm 2 unrank($\mathscr{A} * \mathscr{B}$, n, i)

$c \leftarrow 0; j \leftarrow 0; d \leftarrow count(\mathscr{A}, j) * count(\mathscr{B}, n-j)$
while $i < c + d$ **do**
    $c \leftarrow c + d; j \leftarrow j + 1$
    $d \leftarrow \binom{n}{j} * count(\mathscr{A}, j) * count(\mathscr{B}, n-j)$
end while
$i' \leftarrow i - c$
$l \leftarrow i' mod \binom{n}{j}$ # $l =$ label rank
$i'' \leftarrow i' div \binom{n}{j}; b \leftarrow count(\mathscr{B}, n-j)$
$\alpha \leftarrow unrank(\mathscr{A}, j, i'' div\, b)$
$\beta \leftarrow unrank(\mathscr{B}, n-j, i'' mod\, b)$
return $<< \alpha, \beta >, l >$

Algorithme : ordre lexicographique (Corrigée)

Algorithm 3 unrank($\mathscr{A} * \mathscr{B}$, n, i)

$c \leftarrow 0; j \leftarrow 0; d \leftarrow count(\mathscr{A}, j) * count(\mathscr{B}, n-j)$
while $i \geq c + d$ **do**
  $c \leftarrow c + d; j \leftarrow j + 1$
  $d \leftarrow \binom{n}{j} * count(\mathscr{A}, j) * count(\mathscr{B}, n-j)$
end while
$i' \leftarrow i - c$
$l \leftarrow i' mod \binom{n}{j}$ $\#$ $l =$ label rank
$i'' \leftarrow i' div \binom{n}{j}; b \leftarrow count(\mathscr{B}, n-j)$
$\alpha \leftarrow unrank(\mathscr{A}, j, i'' div\, b)$
$\beta \leftarrow unrank(\mathscr{B}, n-j, i'' mod\, b)$
return $<< \alpha, \beta >, l >$

Algorithme : ordre lexicographique

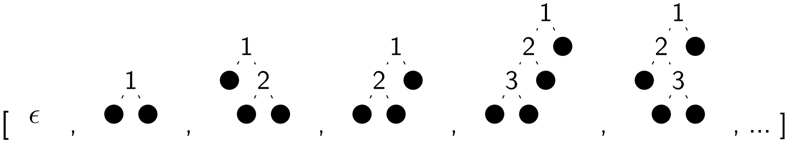
Algorithm 4 unrank($\mathscr{A} * \mathscr{B}$, n, i)

1: **if** $i < |\mathscr{A}_0\mathscr{B}_n|$ **then**
2: **return** i-ème élément de $\mathscr{A}_0\mathscr{B}_n$
3: **end if**
4: $c_j = \sum_{d=0}^{j} |\mathscr{A}_d\mathscr{B}_{n-d}|$
5: **return** $(i - c_j)$-ème élément de $\mathscr{A}_{j+1}\mathscr{B}_{n-(j+1)}$

$i \in [0, ..., |C_n| - 1]$
et $\exists j \in [0, 1, ..., n-1]$ tel que $c_j \leq i < c_{j+1}$.

Exemple

Soit $\mathscr{A} = \mathscr{B} =$

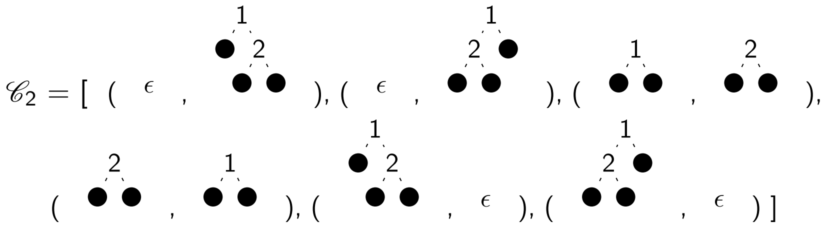


Deux structures d'arbres binaires avec étiquetage croissant.
$\mathscr{A}_n$ et $\mathscr{B}_n$ représentent les arbres binaires ayant n nœuds internes étiquetés.

Exemple 1/2 : Produit

Soit $\mathscr{C} = \mathscr{A} * \mathscr{B}$.

$$\mathscr{C}_2 = \mathscr{A}_0 \star \mathscr{B}_2 + \mathscr{A}_1 \star \mathscr{B}_1 + \mathscr{A}_2 \star \mathscr{B}_0$$



Exemple 2/2 : Produit

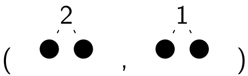
unrank($\mathscr{A} \star \mathscr{B}$, 2, i)

count($\mathscr{A}|\mathscr{B}, 0$)=1 | count($\mathscr{A}|\mathscr{B}, 1$)=1 | count($\mathscr{A}|\mathscr{B}, 2$)=2

$$
\begin{array}{ccccccc}
|\mathscr{C}_2| = & |\mathscr{A}_0\mathscr{B}_2| & + & |\mathscr{A}_1\mathscr{B}_1| & + & |\mathscr{A}_2\mathscr{B}_0| \\
& \uparrow & & \uparrow & & \uparrow \\
& i < 2 & & 2 \leq i < 2+2 & & 2+2 \leq i < 2+2+2
\end{array}
$$

Soit $i = 3$, $2 \leq i < 4$

unrank($\mathscr{A} \star \mathscr{B}$, 2, 3) retourne donc le (3-2=1)-ième élément de $\mathscr{A}_1\mathscr{B}_1 =$



Produit en boîte

Dans le produit en boîte, le plus petit label apparaît uniquement dans le premier élément de la pair.

Exemple avec le produit $\alpha * \beta$, deux objets étiquetés de tailles j et n-j :

- Produit simple : $|\alpha * \beta| = \binom{n}{j}$
- Produit en boîte : $|\alpha * \beta| = \binom{n-1}{j-1}$

Conclusion

- Heuristiques
  - unrank($\mathscr{A} + \mathscr{B}$, n, i) en $O(n^2)$
  - unrank($\mathscr{A} * \mathscr{B}$, n, i) en $O(n^2)$ (**Ordre lexicographique**)
  - unrank($\mathscr{A} * \mathscr{B}$, n, i) en $O(nlog(n))$ (**Ordre boustrophedonic**)
  - unrank($\mathscr{A}^{\square} * \mathscr{B}$, n, i)

Merci pour votre attention
Des questions ?