Unranking of labeled combinatorial classes

Katia A., Firat M.

Sorbonne Université, UFR Ingénierie
Frédéric PESCHANSKI

Novembre 2019

Sommaire

Unranking

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

- son rang.
- sa taille.
- une spécification de $\mathscr{A}$.

Rang de a dans $\mathscr{A}$: nombre d'objets de même taille dans $\mathscr{A}$ qui sont strictement plus petits que a (selon un ordre fixé).
Chaînes de caractères et arbres binaires sont des exemples de structures combinatoires.

3 Problèmes

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

Nous considérons, dans la suite, uniquement le cas des structures combinatoires labellisées.

Spécification

Orderings : 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 $\mathscr{C}$.

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

Orderings : 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.

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

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
```

Produit (ordre lexicographique)

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

$c \leftarrow 0; d \leftarrow count(\mathscr{A}, j) * count(\mathscr{B}, n - j)$
while ss **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
$i' \leftarrow i - c$
$l \leftarrow i' mod \binom{n}{j}$
$i'' \leftarrow i' div \binom{n}{j}; b \leftarrow count(\mathscr{B}, n - j)$
$\alpha \leftarrow unrank(\mathscr{A}, j, i'' div\, b)$
$\beta \leftarrow unrank(\mathscr{B}, n - j, i'' mod\, b)$
return $< (\alpha, \beta), l >$

Produit(ordre boustrophédonique)

la taille du premier composant

Union 1/2

Soit $\mathscr{A}$ et $\mathscr{B}$ des structures combinatoires.
$\mathscr{A}_n$ (resp. $\mathscr{B}_n$) représente une chaîne de caractère composé exactement d'une lettre x (resp. y) et de $n-1$ lettre a (resp. b).
Le i-ème élément de $\mathscr{A}_n$ est la chaîne de caractère ayant la lettre x en i-ème position (avec $1 \leq i \leq n$).

Exemple : $\mathscr{A}_4 = ["xaaa", "axaa", "aaxa", "aaax"]$,
$\mathscr{B}_4 = ["ybbb", "bybb", "bbyb", "bbby"]$

Union 2/2

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

$\mathscr{C}_4$=["*xaaa*","*axaa*","*aaxa*","*aaax*","*ybbb*","*bybb*","*bbyb*","*bbby*"]

unrank($\mathscr{A} \cup \mathscr{B}$, 4, 2) = unrank($\mathscr{A}$, 4, 2) = "*axaa*"

unrank($\mathscr{A} \cup \mathscr{B}$, 4, 7) = unrank($\mathscr{B}$, 4, 3 $(= 7 - 4)$) = "*bbyb*"