Déclasser dans les douze chemins (2025)

Corrigé
*

On étudie, compte et range des classes d'équivalence de fonctions d'un ensemble fini N à n éléments dans un ensemble fini X à x éléments. On s'intéresse à quatre relations d'équivalence pour les fonctions :

— l'égalité, notée DD ;

— l'égalité à permutation des éléments de N près, notée UD ;

— l'égalité à permutation des éléments de X près, notée DU ;

— l'égalité à permutation des éléments de N et de X près, notée UU.

Lorsqu'on a une égalité à permutation près sur un ensemble, on peut considérer que les éléments de cet ensemble sont « indistinguables » (**U***ndistinguishable*) pour la comparaison des fonctions, ce qui explique les notations U et D.
En outre, on s'intéresse à trois caractéristiques des fonctions qu'on étudie :

— les fonctions *arbitraires* (notées A) ;

— les fonctions *injectives* (notées I) ;

— les fonctions *surjectives* (notées S).

Cela définit 12 (3×4) problèmes de dénombrement, appelés *douze chemins* (*twelvefold way*), ou *douze problèmes de Gian-Carlo Rota*. Par exemple, le problème IDU consiste à dénombrer et décrire les classes d'équivalence de fonctions injectives à permutation de X près.
Quand $j \in \mathbb{N}$, on pourra utiliser $[\![1, j]\!]$ pour désigner $\{i \in \mathbb{N} \mid 1 \leqslant i \leqslant j\}$.

Question 1

Dénombrer les classes des problèmes suivants :

1. ADD
2. IDD
3. IUU
4. SUD

Solution.

1. Il suffit de compter les fonctions de N dans X : pour cela, il suffit d'associer à chaque élément de N un élément de X. On obtient x^n.

2. Pour construire une fonction injective, on choisit successivement les images des éléments de N ; on dispose de x, puis $x-1$, puis $x-2$, puis ..., puis $x-n+1$ choix. Au total, on dénombre donc $x \times (x-1) \times \cdots \times (x-n+1) = \dfrac{x!}{(x-n)!}$ fonctions (et 0 si $n > x$).

3. Pour formaliser : deux fonctions injectives f et g sont équivalentes pour IUU s'il existe σ et τ deux permutations telles que $\sigma \circ f \circ \tau = g$. Soient f et g injectives. On définit alors :

$$\sigma : \begin{pmatrix} X & \to & X \\ e & \mapsto & \begin{cases} g(y) \text{ si } e \text{ peut s'écrire comme } e = f(y) \\ e \text{ sinon} \end{cases} \end{pmatrix}$$

Par injectivités de f et de g, σ est injective, donc c'est bien une permutation ; et par construction, $\sigma \circ f = g$. Donc toutes les fonctions injectives sont équivalentes pour IDU (et, en choisissant $\tau = \mathrm{id}_X$, aussi pour IUU). Seul piège : il faut être sûr qu'il existe *au moins* une fonction injective, ce qui est le cas ssi $n \leqslant x$. Donc il y a 1 classe si $n \leqslant x$, et 0 sinon.

4. Deux fonctions surjectives f et g sont équivalentes pour SUD s'il existe τ une permutation telle que $f \circ \tau = g$. Deux fonctions sont donc équivalentes ssi pour tout $y \in X, |f^{-1}(y)| = |g^{-1}(y)|$ (et par ailleurs, ces cardinaux doivent valoir au moins 1). Construire une classe d'équivalence revient donc à attribuer une valeur à chaque $|f^{-1}(y)|$ de sorte à ce que $|f^{-1}(y)| \geqslant 1$ et $\displaystyle\sum_{y \in X} \left|f^{-1}(y)\right| = n$. En posant $\alpha_y = |f^{-1}(y)| - 1$, on cherche donc à attribuer des valeurs positives aux α_y telles que $\displaystyle\sum_{y \in X} \alpha_y = n - x$. Pour ça, on considère avoir $n - x$ symboles « $*$ » alignés, et à intercaler entre eux $x - 1$ symboles « | » : chaque nombre de $*$ entre deux barres symbolisera les α_y. On cherche alors à compter le nombre de ces chaînes ; elles sont de longueur $n - x + x - 1 = n - 1$, et il suffit d'y choisir les positions des $x - 1$ barres. Donc il y a $\displaystyle\binom{n-1}{x-1} = \binom{n-1}{n-x}$ telles classes.

Question 2

Pourquoi la caractéristique *bijective* n'est-elle pas étudiée ?

Solution.

Pour que la bijectivité ait un sens, il faut que $n = x$; mais alors l'injectivité et la surjectivité sont équivalentes à la bijectivité, donc les étudier tout le temps résout aussi la bijectivité.

Question 3

Les problèmes des douze chemins admettent chacun au moins une représentation naturelle de leurs classes d'équivalence et un *ordre lexicographique* sur ces représentations.
Donner de telles représentations et ordres pour les problèmes de la question 1.

Solution.

1. On représente une fonction par la liste de ces éléments ; l'ordre utilisé est l'ordre lexicographique usuel sur les listes.

2. On peut procéder de même que pour les fonctions arbitraires ; on peut néanmoins faire un peu mieux. Par exemple, supposons $N = [\![0, 3]\!]$ et $X = [\![0, 10]\!]$. On souhaite représenter la fonction $f : 0 \mapsto 4, 1 \mapsto 10, 2 \mapsto 1, 3 \mapsto 5$. On pourrait représenter simplement f comme $[4, 10, 1, 5]$. Seulement, cette représentation n'est pas « quelconque », car il faut supposer de plus que la liste est elle-même injective ; il existe des listes qui ne représentent pas des fonctions injectives. On va réduire l'espace des listes manipulées en codant pas *l'image elle-même*, mais *son rang après avoir retiré les images précédentes*. Toujours sur notre exemple, f serait représentée par $[4, 9, 1, 3]$. Le gain est que notre espace d'encodage est plus petit ; une liste quelconque est un élément de $[\![0, 10]\!] \times \cdots \times [\![0, 10]\!]$; ici, c'est un élément de $[\![0, 10]\!] \times [\![0, 9]\!] \times \cdots \times [\![0, 6]\!]$. Aussi, chacun de ces codages correspond bien, par définition, à une fonction injective. L'ordre lexicographique y fonctionne toujours ; mais l'avantage est qu'on peut plus facilement construire un algorithme permettant de passer d'un encodage à une fonction injective.

3. Il y a au plus 1 classe possible, on évacue l'étude de ce cas.

4. En reprenant les notations développées avant, une représentation serait celle des $[\alpha_0, \alpha_1, \ldots, \alpha_{x-1}]$. L'ordre lexicographique marche toujours, mais demandera plus de travail pour la future énumération.

Un *algorithme de déclassement* pour un problème des douze chemins est un algorithme qui prend en entrée un entier i et renvoie en sortie une représentation de la classe d'équivalence située à la $i+1$-ème place selon l'ordre lexicographique associée à ce problème.

Question 4

Donner des algorithmes de déclassement pour les problèmes de la question 1.

Solution.

1. Il suffit d'écrire i en base x-aire. Une méthode par divisions est amplement suffisante.

2. Avec la seconde représentation avec le rang, la question devient tout de suite plus sympathique : cela revient à l'écriture de i non pas en base constante x, mais en base $[x, x-1, \ldots]$. La méthode par divions marche encore.

4. Pour trouver la représentation associée à i, on part de la plus petite, qui est $[0, 0, \ldots, 0, n-x]$. À partir d'une représentation, on passe alors à la suivante dans l'ordre lexicographique selon l'idée suivante : « on considère le compteur tout à droite ; on cherche à le décrémenter, pour pouvoir incrémenter son voisin de gauche ». On a besoin d'une règle supplémentaire :

— si le dernier compteur est nul, on remonte vers la gauche jusqu'à un compteur c non nul ; alors c se sépare en deux, en donnant 1 à son voisin de gauche, et le reste au dernier compteur (et c devient nul).

Dans la suite, on s'intéresse uniquement au problème SDU.

Question 5

Donner une relation de récurrence décrivant le dénombrement des fonctions de SDU.

Solution.

SDU signifie que f et g sont équivalentes si $\sigma \circ f = g$. Cela signifie que si f envoie deux éléments sur une même image, g aussi.
Soit $SDU(n, x)$ le nombre de classes de SDU.
Alors, si $n > 0$, $SDU(n, 0) = 0$: je n'ai pas de fonction vers l'ensemble vide (donc aucune surjective). Si $x > 0$, $SDU(0, x) = 0$: je n'ai pas de fonction surjective depuis l'ensemble vide. Enfin, $SDU(n, n) = 1$ (même avec $n = 0$).
Ensuite, considérons $SDU(n, x+1)$; considérons alors le $x+1$-ème élément de X. J'ai alors les possibilités suivantes pour choisir son ensemble antécédent :

— soit je décide qu'il sera seul. Alors pour compléter ma fonction, je dois choisir un élément de $SDU(n-1, x)$;

— soit je décide qu'il aura des amis. Je choisis donc un élément de $SDU(n, x)$, et il me reste n possibilités pour le choix de l'antécédent du $x+1$-ème élément de X.

On obtient :

$$SDU(n, x+1) = SDU(n-1, x) + n \times SDU(n, x)$$

Soit A et B deux ensembles d'entiers naturels, on définit $A \prec B = (\min A \leqslant \min B)$. On notera $A \backslash B$ pour $\{x \in A \mid x \notin B\}$. On définit enfin :

$$A \lhd B = \begin{cases} & A = B \\ \text{ou} & A \subset B \text{ et } \max A < \min(B \backslash A) \\ \text{ou} & B \subset A \text{ et } \min(A \backslash B) < \max B \\ \text{ou} & \min(A \backslash B) < \min(B \backslash A) \end{cases}$$

Question 6

Donner une représentation des classes de SDU, puis un ordre total sur ces classes.

Solution.

En fait, pour représenter un élément de SDU, il suffit de se demander : « quels éléments de N ont les mêmes images ? ». Cela suffit à caractériser un élément de SDU. Autrement écrit, un élément de SDU est une relation d'équivalence dans N en x classes ! On représente alors un élément de SDU comme une liste d'ensembles formant une partition de N qui est triée pour l'ordre $\prec$. Par exemple, soit $f : [\![0,5]\!] \to [\![0,2]\!]$ avec $f(0) = f(2) = f(5) = 0$, $f(3) = 1$ et $f(1) = f(4) = 2$. Alors on représente f par :

$$[[0,2,5],[1,4],[3]]$$

On se propose alors de fixer, sur ces représentations, l'ordre lexicographique introduit par $\lhd$. Pour justifier que cet ordre est total, il faut montrer que $\lhd$ est un ordre total sur $\mathcal{P}(\mathbb{N})$. Cette fois, on représente un élément A de $\mathcal{P}(\mathbb{N})$ comme la liste triée $L(A)$ de ses éléments : on se rend alors compte que $A \lhd B$ ssi $L(A) \leqslant_{\text{lex}} L(B)$ (où $\leqslant_{\text{lex}}$ désigne l'ordre lexicographique généré par l'ordre usuel sur $\mathbb{N}$). $\leqslant$ est total, donc $\leqslant_{\text{lex}}$ est total ; donc $\lhd_{\text{lex}}$ est total. Ouf !

Question 7

Donner les 10 premières fonctions, pour cet ordre, de SDU avec $n = 5$ et $x = 3$.

Solution.

1	2	3,4,5
1	2,3	4,5
1	2,3,4	5
1	2,3,5	4
1	2,4	3,5

1	2,4,5	4
1	2,5	3,4
1,2	3	4,5
1,2	3,4	5
1,2	3,5	4

Question 8

Donner un algorithme de déclassement pour SDU.

Solution.

On part de la représentation la plus petite, et on cherche à trouver la représentation suivante. L'idée générale est « on prend l'élément le plus petit de la dernière case d, et on le rajoute à l'avant-dernière case ». Règle supplémentaire :

— si d contient 1 seul élément, on considère c la case la plus à droite qui contient plusieurs éléments. On cherche alors son successeur sur l'ordre $\lhd$ (ce qui correspond à l'ordre $\leqslant_{\text{lex}}$) (en n'utilisant que des éléments présents dans les cases à droite de c comprise). On répartit ensuite les éléments restants sur les cases à droite de c : on dépose un par un ces éléments, les plus petits d'abord, un par case, jusqu'à arriver à la dernière case, dans laquelle on met tout le reste.