Progrès dans l'énumération des posets

C. Chaunier, N. Lygeros

Résumé - On décrit l'algorithme qui a permis de calculer le nombre de posets à isomorphie prés ayant 12 éléments : $P_{12} = 1104891746$ [1], qui constitue actuellement la plus grande valeur connue dans cette énumération.

Progress in the enumeration of posets

Abstract - We describe what algorithm has enable to compute the number of nonisomorphic posets on 12 elements : $P_{12} = 1,104,891,746$, which currently constitutes the greatest known value of this enumeration.

Introduction. - Dans cette Note, qui fait suite à [1], nous ne considérons que des posets non isomorphes 2 à 2. On note P_n le nombre de posets à n éléments. Le calcul exact de P_n est un problème très difficile et donc actuellement ouvert (voir [2]), Ainsi l'utilisation des ordinateurs est rendue nécessaire. Les recherches dans ce domaine ont commencé il y a une vingtaine d'années par l'obtention à la "main" de la première valeur non triviale P_7 par Wright [3]. Le premier calcul sur ordinateur, effectué par S.K. Das [4], donne P_8. Pour calculer P_9, Möhring [5] s'appuie sur l'identification des graphes de comparabilité en exploitant les données de R.C. Read et N.C. Wormald [6] sur les graphes. Enfin pour parvenir à P_{10} et P_{11}, J.C. Culberson et G. J. E. Rawlins [7] élaborent le premier algorithme spécifique aux posets en s'appuyant sur la structure du poset des posets à n éléments. Quant à notre algorithme pour obtenir P_{12}, il est (entre autres) fondé sur une idée qui tire son origine heuristique de la théorie des fractals [8].

On a le tableau 1 :

TABLEAU I

$0 \leq n \leq 6$, $P_n = 1; 1; 2; 5; 16; 63; 318;$

$P_7 =$	2 045	(1972)	J. Wright
$P_8 =$	16 999	(1977)	S. K. Das
$P_9 =$	183 231	(1984)	R. H. Möhring
$P_{10} =$	2 567 284	(1990)	J. C. Culberson et G. J. E. Rawlins
$P_{11} =$	46 749 427	(1990)	J. C. Culberson et G. J. E. Rawlins
$P_{12} =$	1 104 891 746	(1991)	

STRUCTURE DE L'ALGORITHME. - L'algorithme énumère les posets à n sommets en les générant tous. Pour cela on représente un poset $P = (\{x_1, \ldots, x_n\}, <)$ par sa matrice d'incidence $(a_{ij}) \in \{0,1\}^{n^2}$ avec $a_{ij} = 1$ si et seulement si $x_i < x_j$. On améliore alors l'algorithme naïf consistant à générer toutes les matrices transitives et à éliminer celles qu'une permutation simultanée $(a_{\sigma(i)\sigma(j)})$ des lignes et colonnes rendrait égale à une matrice déjà produite. On privilégie un premier ordre sur les lignes et colonnes qui réduit considérablement l'ensemble des permutations à vérifier ainsi que la redondance parmi les matrices générées, en exigeant la

Condition 1. - La suite $(d_i)_{1 \leq i \leq n}$ est décroissante, où $d_i = \sum_{j=1}^{n} a_{ij}$
Condition qui implique une proposition à rapprocher sans doute du lemme d'anticircularité de G. Birkhoff [9] :

Proposition 1. - (a_{ij}) est plus que triangulaire supérieure, i.e. $\forall k, l / d_k = d_{k+l-1}$ les $l(l+1)/2$ coefficients $a_{ij}, k \leq i \leq j < k+l$ sur et au-dessus de la diagonale, sont nuls. La condition suivante permet d'identifier les

1. Une annonce de ce résultat a été envoyée aux 'Abstracts de l'A.M.S.' le 3.10.1991.

permutations aux seuls automorphismes, suivant une méthode introduite par R.C. Read [10] :

Condition 2. - $(a_{1,n}, a_{1,n-1}, \ldots, a_{1,1}, a_{2,n}, \ldots, a_{n,1})$ est maximal pour l'ordre lexico-graphique de $\{0,1\}^{n^2}$ en attribuant le plus grand poids à $a_{1,n}$.

Elle rend inutile la comparaison de chaque matrice produite avec les précédentes, et donc leur mémorisation : Il suffit d'éprouver sur la matrice qu'aucune permutation n'augmente le vecteur ci-dessus pour la compter. L'utilisation d'un backtracking pour ce test permet de gagner du temps, notamment sur les permutations diminuant le vecteur. Lorsque le vecteur est effectivement maximal, le test donne au passage tous les automorphismes du poset, ce qui s'avère utile dans une étude de la structure des posets plutôt que de leur seule quantité.
La conjonction des deux conditions permet de réduire encore la redondance de la génération. Elle entraîne ainsi la proposition suivante qui doit sa genèse à une notion appartenant à la théorie des fractals, l'autosimilarité :

Proposition 2. - Pour tout i, soit A_i la partition des indices $\{1, 2, \ldots, n\}$ réunissant j et j' dans une même classe si et seulement si $d_j = d_{j'}$ et $\forall k < i, a_{kj} = a_{kj'}$. Alors $j < j'$ dans une même classe de A_i, on a $a_{ij} \leq a_{ij'}$.

Pour avoir connaissance des partitions A_j et limiter les valeurs possibles des lignes i correspondantes, il est donc avantageux de se donner la suite d_i avant de générer, par ordre croissant des lignes, les matrices qui possèdent cette suite de degrés. Par ailleurs l'ordre croissant des lignes simplifie la condition de transitivité, qui devient pour chaque ligne i :

$$i < j < k, a_{ij} = 1 \text{ et } a_{ik} = 0 \Rightarrow a_{jk} = 0$$

Connaissant d_j on peut également éprouver que le nombre de coefficients nuls d'une ligne j ne dépasse pas $n - d_j$. Le fait que la génération d'une ligne dépende des précédentes justifie l'usage d'un backtracking d'indice i également dans la partie génératrice de l'algorithme.
Quand la dernière ligne est atteinte, les conditions 1 et 2 améliorent aussi le test validant la matrice obtenue :

Proposition 3. - Pour éprouver la condition 2, l'indice i peut n'être permuté qu'avec les indices de sa classe dans $A_i, \forall i$.
L'efficacité est due au fait que la plupart des classes considérées ne contiennent qu'un sommet, à rapprocher de la rigidité de presque tous les posets [11]. Pour réduire les autres on utilise :

Proposition 3 bis. - Il est inutile de permuter l'indice i avec les indices k de sa classe dans A_i, qui ont des lignes identiques $(a_{ij})_{1\leq j\leq n} = (a_{kj})_{1\leq j\leq n}$.

Les deux conditions impliquent aussi une proposition déjà énoncée par C. J. Colbourn [12] et qui infiltre la génération des posets de la même façon que les propositions 1 et 2 :

Proposition 4. - $\forall i < k$ dans la même classe de A_i, leur ligne vérifie pour l'ordre lexicographique :

$$(a_{i,n}, a_{i,n-1}, \ldots, a_{i,1}) \geq (a_{k,n}, a_{k,n-1}, \ldots, a_{k,1})$$

Comparaison. L'algorithme se présente donc ainsi :
- On décide de générer tous les posets dans un ordre quelconque, avec une génération superflue globale, élaguée ponctuellement par le test de canonicité (condition 2).
-On améliore cet élagage par la réduction du nombre des permutations à considérer (condition 1. propositions 3 et 3 bis).
-On déduit la génération superflue globale par une infiltration globale du test de canonicité dans la génération (propositions 1, 2 et 4).
Par comparaison, pour l'algorithme de Culberson et Rawlins :
-On décide de parcourir en profondeur d'abord, le poset des posets non étiquetés, avec une génération superflue locale, mais élaguée non localement.
-On améliore cet élagage par l'affaiblissement du test d'isomorphie qui devient un test d'abritement, ce qui implique une réduction de l'ensemble des posets à comparer au poset nouvellement produit.

-On déduit la génération superflue locale par une infiltration globale du test d'abritement dans la génération.

RESULTATS. -On obtient le tableau suivant, énumérant les posets à 12 éléments suivant la relation r.

TABLEAU II

r	P^r_{12}	r	P^r_{12}	r	P^r_{12}
0	1				
1	1	23	25 468 042	45	6 370 240
2	3	24	33 157 695	46	4 491 015
3	7	25	41 495 336	47	3 100 063
4	19	26	50 008 606	48	2 094 942
5	47	27	58 130 096	49	1 386 092
6	133	28	65 270 723	50	897 535
7	352	29	70 888 253	51	568 627
8	997	30	74 562 234	52	352 196
9	2 753	31	76 042 383	53	213 115
10	7 558	32	75 275 671	54	125 818
11	19 801	33	72 402 491	55	72 382
12	49 795	34	67 726 046	56	40 515
13	117 875	35	61 666 534	57	21 985
14	263 019	36	54 699 028	58	11 545
15	550 013	37	47 302 979	59	5 808
16	1 080 422	38	39 908 316	60	2 779
17	1 993 865	39	32 869 931	61	1 249
18	3 469 819	40	26 443 898	62	509
19	5 707 944	41	20 792 175	63	184
20	8 909 624	42	15 984 309	64	55
21	13 234 277	43	12 020 498	65	11
22	18 766 663	44	8 844 848	66	1

Remarques. Les valeurs pour $0 \leq r \leq 7$ coïncident avec celles obtenues par Culberson et Rawlins.
(ii) Les valeurs pour $55 \leq r \leq 66$ sont confirmées par les formules de M. Erné [6].
-La multiplicité des machines utilisées ne permet de donner qu'une estimation du temps mis par notre implémentation de l'algorithme : 14 jours si on l'effectuait sur un micro-processeur 80486 à 25 MHz.
-La mémoire utilisée est asymptotiquement de n3 hits, nécessités seulement par l'empilement des conditions de transitivité lors du backtracking.
Un gain de temps de 10pc est réalisé avec le langage C si l'on donne une puissance de 2 comme taille aux seconds indices des tableaux.
-Le fait que l'algorithme commence par définir la suite (d_i) permet un partitionnement fin des calculs.

Nous remercions R. Chaunier pour son aide constante, R. Fraïssé qui a été l'initiateur de ce projet, et surtout M. Pouzet sans qui P_{12} n'aurait jamais été obtenu.
Ce travail a été finance par le P.R.C. Math. et Info. C.N.R.S.
Note remise le 2 novembre 1991, acceptée après révision le 11 mars 1992.

Références bibliographiques

[1] R. Fraïssé et N. Lygeros. Petits posets : dénombrement, représentabilité par cercles et "compenseurs". *C. R. Acad. Sci. Paris.* 313, série I. 1991. p. 417-420.
[2] R. P. Stanley, *Enumerative Combinatorics.* 1, Wadsworth et Brooks col. math. Ser,. California. 1986.
[3] J. Wright, Cycle indicators of certain classes of types of quasi-orders or topologies. *Ph. Dissertation*, U. of Rochester, 1972.
[4] S. K. Das. A machine representation of finite T_0 topologies. *J. Assoc. Comp. Machinery*, 24. N° 4. 1977. p. 676-692.

[5] R.H. Möhring. Algorithmic aspects of comparability graphs and interval graphs. in : *Graph and Order*, I. Rival, éd. (Reidel, Dordrecht, 1985) 41-101.
[6] R.C. Read et N.C. Wormald, Catalogues of graphs and digraphs (announcement). *Discrete Math.* 31 (1980) 224.
[7] J.C Culberson et J.E. Rawlins. New results from an algorithm for counting posets, *Order 7* (1991) 361-374.
[8] B. Mandelbrot. *Les objets fractals*, *Flammarion*, 1ère édn. (1975).
[9] G Birkhoff, *Lattice theory. A.M.S. colloque publ.* vol. 25. 3° édition (1967).
[10] R.C. Read. Every one a winner, *Ann. Discrete Math. 2* (1978) 107-120.
[11] H.J. Prömel, Counting unlabeled structures. *J. Combin. Theory.* Series A 44 (1987) 83 93.
[12] C J. Colbourn. *Graph generation.* Research Report CS-77-37. University of Waterloo. 1977. p. 100.
[13] M. Erné, The number of partially ordered sets with more points that incomparable pairs, *Discrete Math.* (preprint)