Projet de parallélisme

Calcul distribué des polynômes de spin des tableaux de rubans

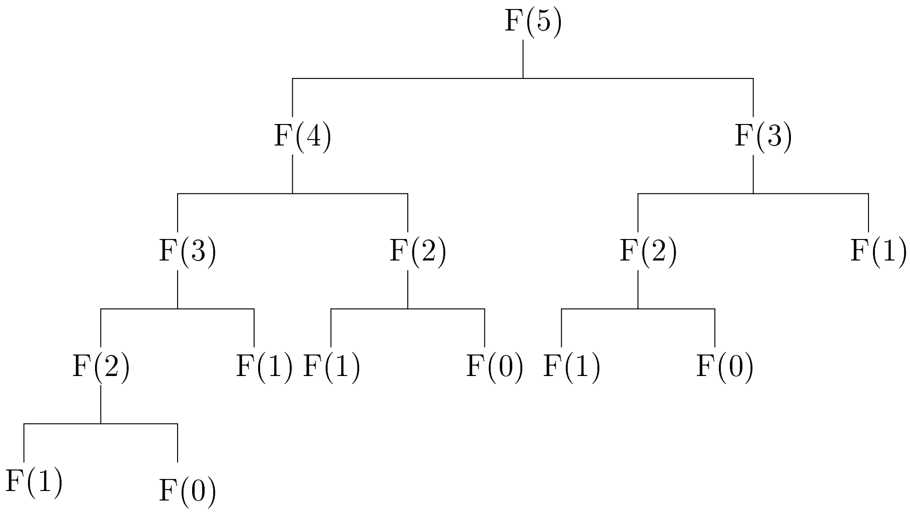
1 Introduction

Le but du projet est de mettre en place le calcul distribué d'une fonction récursive au moyen de l'environnement PVM [2]. Ce texte présente une application importante, issue de recherches en cours, mais on recommande de commencer par tester les protocoles sur un exemple trivial, par exemple la suite de Fibonacci (définie par $F(n) = F(n-1)+F(n-2)$ pour $n \geq 2$, et $F(0) = 0, F(1) = 1$).

Le principe est le suivant. On veut calculer la valeur d'une certaine fonction F sur un élément x d'un certain ensemble ordonné X. On suppose que toute chaîne strictement décroissante $x_1 > x_2 > \ldots > x_r > \ldots$ d'éléments de X est finie, et que F est définie de manière récursive : les valeurs de F sur les éléments minimaux de X sont connues, et on sait, pour tout x non minimal, exprimer $F(x)$ en fonction d'un nombre fini d'éléments $x_1, x_2, \ldots$ strictement inférieurs à x. Le nombre de ces éléments peut dépendre du choix de x.

Si l'élément x_i n'est pas minimal, le calcul de $F(x_i)$ se fera en exprimant $F(x_i)$ en fonction de $F(x_{i,1})$, $F(x_{i,2})$, et ainsi de suite. Le calcul de $F(x)$ peut donc être représenté par un arbre, dont les noeuds sont étiquetés par les éléments $x_{i_1 i_2 \ldots i_k}$ sur lesquels on doit évaluer F, et contiennent dans le cas général une information supplémentaire indiquant comment utiliser la valeur de $F(x_{i_1 i_2 \ldots i_k})$, une fois celle-ci calculée.

Bien entendu, il arrivera fréquemment que plusieurs sommets d'un même niveau possèdent la même étiquette, et il sera inutile de recalculer plusieurs fois la valeur correspondante. Il faut donc gérer une table des valeurs déjà calculées ou dont le calcul a déjà été demandé. Par exemple, l'arbre associé au calcul de $F(5)$ est



2 Distribution du calcul

On propose de simuler un réseau en arbre. Chaque noeud de l'arbre d'évaluation sera géré par un processeur (virtuel). La racine gérera la table des valeurs calculées ou en cours de calcul, et collectera les résultats des noeuds intermédiaires. Lorsque toutes les valeurs intermédiaires seront calculées, la racine calculera le résultat final. Le noeud chargé de calculer $F(z)$ commence par calculer les éléments $z_1, z_2, \ldots, z_n$ sur lesquels il doit évaluer F, et en communique la liste à la racine. S'il y a lieu, il communique aussi pour chacun d'eux les paramètres nécéssaires à l'intégration de la valeur calculée dans le résultat final. La racine compare les z_i avec les valeurs stockées dans sa table, et renvoie une autorisation de calcul pour celles qui n'y figuraient pas. Elle met ensuite la table à jour, en inscrivant les numéros des processeurs chargés du calcul de ces valeurs (ces numéros lui sont transmis par le noeud qui a activé les processeurs correspondants). Si toutes les valeurs transmises par un noeud à la racine figuraient déjà dans la table, le processus associé à ce noeud se termine.

Pour des exemples de taille réaliste, on ne disposera pas d'un nombre suffisant de processeurs, ni même, pour la simulation, d'un nombre suffisant de processus sur une même machine. Il faudra dans ce cas prévoir un programme séquentiel pour le calcul de F, que l'on fera exécuter par les processeurs qui n'auront pas la possibilité de créer les noeuds suivants.

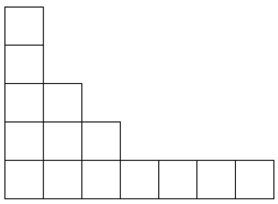
3 Un exemple: les tableaux de rubans

Certains travaux récents de Combinatoire Algébrique [4, 1] mettent en évidence l'intérêt de structures planaires appelées *tableaux de rubans*, qui généralisent les *tableaux de Young* (pour ces derniers, on pourra consulter [3]). Les calculs de tableaux de rubans sont très coûteux en mémoire, comme la plupart des calculs combinatoires de ce type.

3.1 Partitions et tableaux

Une *partition* est une suite finie $\lambda = (\lambda_1 \geq \lambda_2 \geq \ldots \lambda_r \geq 0)$ d'entiers ≥ 0, décroissante au sens large. Les entiers λ_i s'appelent les *parts* de λ. Leur somme n est le *poids* de λ, noté $n = |\lambda|$, et le nombre de parts non nulles est la *longueur* de λ, notée $\ell(\lambda)$.

On peut représenter graphiquement une partition au moyen d'un diagramme de boîtes dans le plan, appelé *diagramme de Ferrers*. La i-ème ligne de ce diagramme est formée de λ_i boîtes, et les extrémités gauches des lignes sont superposées. On numérote les lignes de bas en haut. Par exemple, la partition $\lambda = (7, 3, 2, 1, 1)$ correspond au diagramme



Si on lit maintenant les longueurs des colonnes au lieu de celles des lignes, on obtient une nouvelle partition λ' dite *conjuguée* de λ. Ici, $\lambda' = (5, 3, 2, 1, 1, 1, 1)$.

Un remplissage du diagramme de Ferrers d'une partition λ avec des entiers s'appelle un *tableau de Young* si les deux conditions suivantes sont vérifiées :

1. chaque ligne lue de gauche à droite est croissante au sens large

2. chaque colonne lue de bas en haut est strictement croissante

Si on a utilisé μ_1 fois le nombre 1, μ_2 fois le nombre 2 *etc.*, on dit que le tableau est de *poids* $\mu = (\mu_1, \mu_2, \ldots)$ Par exemple,

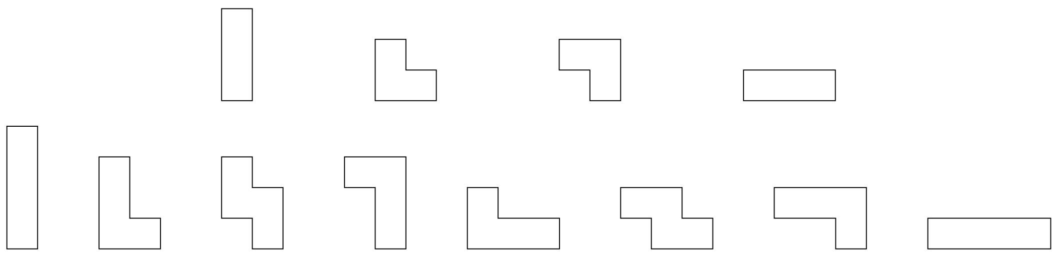
8						
6						
3	5					
2	2	6				
1	1	1	2	4	4	7

est un tableau de forme $(7, 3, 2, 1, 1)$ et de poids $(3, 3, 1, 2, 1, 2, 1, 1)$.

3.2 Tableaux de rubans

On s'intéresse maintenant aux pavages des diagrammes de Ferrers par des *rubans*. Un k-ruban est un diagramme de boîtes *connexe* (on peut aller de toute boîte à toute autre par un chemin de boîtes ayant un côté commun), qui se dirige toujours horizontalement ou en descendant quand on le parcourt de gauche à droite, et qui ne contient pas de carré 2×2. On vérifie facilement que pour k fixé, il y a exactement 2^{k-1} formes de k-rubans. Pour $k = 2$ ce sont les deux dominos

et pour $k = 3, 4$ on a les formes suivantes



Tous les diagrammes de Ferrers ne sont pas pavables par des k-rubans. Une condition nécéssaire évidente est que le nombre de boîtes soit multiple de k. Mais cette condition n'est pas suffisante. Par exemple, la forme $(3, 2, 1)$ n'est pas pavable par des dominos.

Ayant pavé une forme λ par des k-rubans, un remplissage de ce pavage par des entiers sera un *tableau de k-rubans* si comme précédemment, il y a croissance large dans les lignes et croissance stricte dans les colonnes. Cette définition n'est pas suffisament précise, car il faut expliquer comment on lit les lignes et les colonnes. Appelons *racine* d'un ruban sa boîte qui est le plus en bas à droite. La condition précise sur les colonnes est que la racine d'un ruban numéroté i ne doit jamais se trouver au dessus d'une case quelconque d'un ruban portant un numéro inférieur ou égal. On définit le poids d'un tableau de rubans comme dans le cas des tableaux ordinaires.

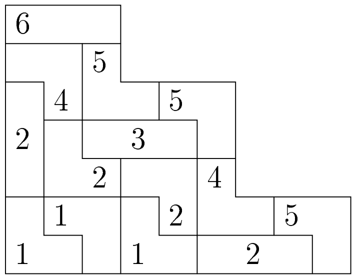


Figure 1: un tableau de rubans de forme (9966633) et de poids (341231)

3.3 Le problème

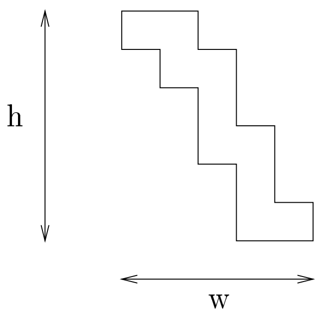
On donne deux partitions λ, μ. On suppose que $|\lambda| = k|\mu|$, et on demande d'engendrer tous les tableaux de rubans de forme λ et de poids μ. Il faudra en plus calculer pour chacun de ces tableaux T une certaine statistique appelée *spin* et notée $s(T)$. On voudrait également être capable de calculer seulement les *polynômes de spin*

$$G^{(k)}_{\lambda\mu}(q) = \sum_{T \in \mathrm{Tab}_k(\lambda,\mu)} q^{s(T)}$$

où $\mathrm{Tab}_k(\lambda, \mu)$ désigne l'ensemble des tableaux de k-rubans de forme λ et de poids μ, sans pour autant engendrer tous les tableaux, qui sont souvent très nombreux.

Le *spin* d'un ruban R est par définition $s(R) = (h(R) - 1)/2$, où $h(R)$ est la *hauteur* de R. On remarque que si w est la *largeur* de R, on a pour un k-ruban R

$$h + w = k + 1 \ . \tag{1}$$

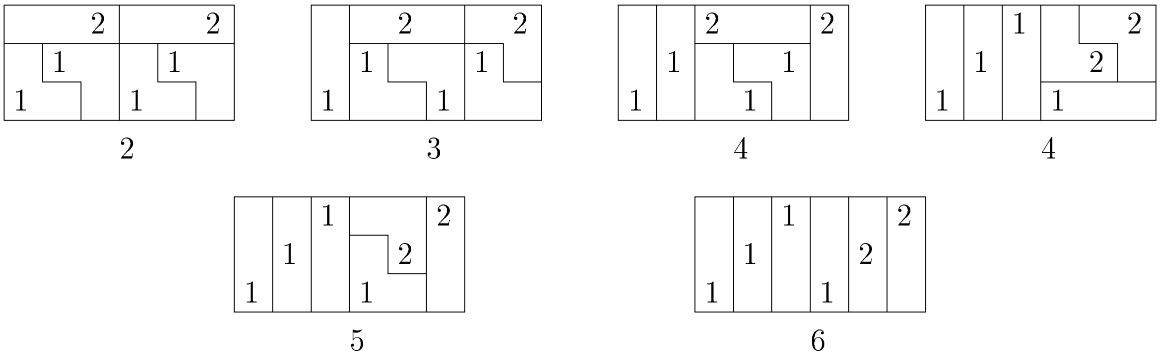


Le spin d'un tableau de rubans T est maintenant défini par

$$s(T) = \sum_R s(R) \tag{2}$$

où la somme porte sur tous les rubans qui composent T. Par exemple, le tableau de la figure 1 a pour spin 6.

Exemple Les tableaux de 3-rubans de forme $(6, 6, 6)$ et de poids $(4, 2)$ sont les suivants (on a indiqué en dessous de chaque tableau son spin)



de sorte que

$$G^{(3)}_{666,42}(q) = q^2 + q^3 + 2q^4 + q^5 + q^6 \ .$$

3.4 Codage matriciel des tableaux de rubans

Un tableau de rubans T de forme λ et de poids μ sera codé par une matrice $M(T)$ à r lignes et s colonnes, r étant la longueur de μ et s celle de la partition conjuguée λ'. Les éléments de cette matrice seront tous égaux à 0 ou à k, et la i-ème ligne (numérotée à partir de celle du bas) contiendra le nombre k exactement μ_i fois. Par exemple, la matrice $M(T)$ associée au tableau de 3-rubans de la figure 1 sera

$$\begin{bmatrix} 0&0&3&0&0&0&0&0&0\\ 0&0&0&3&0&3&0&0&3\\ 0&3&0&0&0&0&3&0&0\\ 0&0&0&0&3&0&0&0&0\\ 3&0&3&0&3&0&0&3&0\\ 0&3&3&0&3&0&0&0&0 \end{bmatrix}$$

Pour obtenir la i-ème ligne de cette matrice, on regarde les numéros des colonnes dans lesquelles se trouvent les racines des rubans numérotés i, et on écrit des k dans les colonnes correspondantes de la matrice. Les autres entrées sont nulles. Sur l'exemple, les rubans numérotés 2 ont leurs racines dans les colonnes $1, 3, 5, 8$. La deuxième ligne (en partant du bas) de la matrice sera donc [303030030].

Pour reconstruire le tableau à partir de la matrice, on commence par placer les rubans numérotés 1. La ligne du bas de la matrice donne les emplacements des racines, et on doit *replier* les rubans. On effectue cette opération de la manière suivante. Si la matrice a n colonnes, on ajoute à sa premiere ligne le vecteur $\rho_n = [n-1, n-2, \ldots, 2, 1, 0]$. Sur notre exemple, on a donc

$$\begin{array}{ccccccccccc} & 0 & 3 & 3 & 0 & 3 & 0 & 0 & 0 & 0\\ + & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0\\ = & 8 & 10 & 9 & 5 & 7 & 3 & 2 & 1 & 0 \end{array}$$

A ce stade, de deux choses l'une : ou bien les entiers obtenus ne sont pas tous distincts, auquel cas il n'existe aucun tableau de rubans correspondant à la matrice qu'on examine, ou bien ils le sont. Dans ce dernier cas, on les trie en ordre décroissant, et on note le nombre d'inversions I de la permutation qui réalise le tri (c'est le nombre de transpositions effectuées par le tri par bulles). Ici, on obtient la suite triée $[10, 9, 8, 7, 5, 3, 2, 1, 0]$ et $I = 3$. On obtient la forme conjuguée de la forme occupée par les rubans 1 en soustrayant ρ_n à la suite triée, et le nombre d'inversions I est égal à $\sum_R (w(R)-1)$, nombre à partir duquel on peut reconstituer le spin: $2s = n(k-1)-I$, où n est le nombre de k-rubans. Sur l'exemple, les rubans numérotés 1 sont tous les trois de largeur 2, d'où $I = 3 = (2-1)+(2-1)+(2-1)$. Ils occupent la forme $\mu = (5,4)$ dont la conjuguée est $\mu' = (2,2,2,2,1)$, et on a bien

$$[10,9,8,7,5,3,2,1,0] - [8,7,6,5,4,3,2,1,0] = [2,2,2,1,1,0,0,0,0] \ .$$

Pour construire la forme occupée par les rubans numérotés 2, on part de la suite triée obtenue à l'étape précédente, et on lui ajoute la ligne de la matrice correspondant aux 2. On procède exactement comme à l'étape 1 : on trie la suite obtenue en ordre décroissant, en ajoutant à la variable I le nombre d'inversions de la permutation réalisant le tri. Si la suite obtenue a des répétitions, c'est que la matrice ne provenait pas du codage d'un tableau de rubans. Sinon, en soustrayant ρ_n à la suite triée, on obtient la forme conjuguée de la forme occupée par les 1 et les 2. Comme on connaissait déja celle occupée par les 1, on en déduit celle occupée par les 2. La variable I est maintenant égale à $\sum_R (w(R)-1)$, où la somme porte cette fois sur les rubans numérotés 1 et 2. On place ensuite les 3 de la même manière, et on itère le procédé jusqu'à ce que tous les rubans soient placés. Sur l'exemple, le placement des 2 correspond à l'opération

$$\begin{array}{ccccccccccc} & 10 & 9 & 8 & 7 & 5 & 3 & 2 & 1 & 0 \\ + & 3 & 0 & 3 & 0 & 3 & 0 & 0 & 3 & 0 \\ = & 13 & 9 & 11 & 7 & 8 & 3 & 2 & 4 & 0 \end{array}$$

La suite triée est $[13,11,9,8,7,4,3,2,0]$, d'où la forme conjuguée $\nu' = [5,4,3,3,3,1,1,1,0]$ de la forme $\nu = (8,5,5,2,1)$ occupée par les 1 et les 2. Le nombre d'inversions de la permutation est 4, donc à ce stade $I = 3+4 = 7$. Le spin du sous-tableau formé par les 1 et les 2 est donc $s = (7 \times (3-1) - I)/2 = \frac{7}{2}$.

4 Calcul récursif des polynômes de spin

Le codage précédent permet de donner une formule récursive pour les polynômes suivants, à partir desquels on calcule facilement les polynômes de spin. Dans la formule qui suit, il faut interpréter la variable α comme $\alpha = \lambda' + \rho_n$, où λ est la forme des tableaux de rubans que l'on veut énumérer, μ leur poids, et $n \geq \lambda_1$:

$$F^{(k)}(\alpha,\mu) = \sum_{T \in \mathrm{Tab}\,_k(\lambda,\mu)} \sum_{R\in T} q^{(w(R)-1)} \qquad (3)$$

Supposons que $\mu = (\mu_1, \ldots, \mu_r)$. On obtient les dispositions possibles des μ_r rubans numérotés r dans un tableau de forme λ et de poids μ en déroulant l'algorithme précédent à l'envers : les formes que peuvent occuper les rubans de numéros $1, \ldots, r-1$ s'obtiennent en soustrayant à α toutes les permutations distinctes du vecteur $v_r = [0, \ldots, 0, k, \ldots, k]$ (μ_1 composantes égales à k, les autres nulles), et en triant en ordre décroissant les vecteurs obtenus, avec conservation du nombre d'inversions. On construit ainsi un arbre, de racine étiquetée α, dont les arêtes sont étiquetées par les permutations distinctes v_r^σ de v_r, et les sommets par les suites $\alpha - v_r^\sigma$ réordonnées, et contiennent aussi le nombre d'inversions de la permutation réordonnante. Les arêtes conduisant à des suites qui contiennent des répétitions sont éliminées. Soient $\alpha_1, \ldots, \alpha_m$ les étiquettes des sommets restants, et $I_1, \ldots, I_m$ les nombres d'inversions correspondants. Posons aussi $\mu^* = (\mu_1, \ldots, \mu_{r-1})$ (on a retiré la dernière part). Alors, la formule de récurrence est

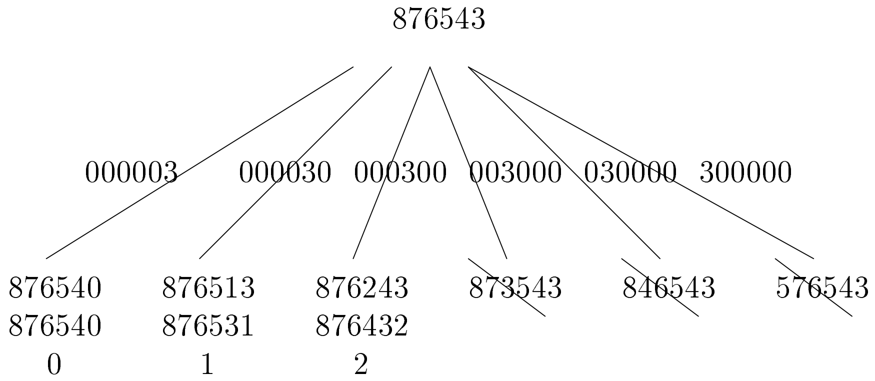
$$F^{(k)}(\alpha, \mu) = \sum_p q^{I_p} F^{(k)}(\alpha_p, \mu^*) \ . \tag{4}$$

Pour que la définition soit complète on précise la condition $F(\beta, [\,]) = 1$ (dans ce cas, $\beta =$ doit être égal à ρ_n sinon c'est qu'une erreur s'est introduite à l'une des étapes précédentes).

Exemple Pour énumérer les tableaux de 3-rubans de forme $\lambda = (6,6,6)$ et de poids $\mu = (4,1,1)$ on calcule $\lambda' = (3,3,3,3,3,3)$. Il faut prendre $n = 6$ (nombre de colonnes). Le vecteur α (racine de l'arbre) est $\alpha = [3,3,3,3,3,3] + [5,4,3,2,1,0] = [8,7,6,5,4,3]$. Ici le nombre de parts de μ est $r = 3$, et le vecteur v_3 est donné par $v_3 = [0,0,0,0,0,3]$. Il a 6 permutations distinctes, parmi lesquelles seulement trois conduisent à des suites admissibles. On a donc

$$F^{(3)}([8,7,6,5,4,3],[4,1,1]) = F^{(3)}([8,7,6,5,4,0],[4,1]) + q\, F^{(3)}([8,7,6,5,3,1],[4,1])$$
$$+ q^2 F^{(3)}([8,7,6,4,3,2],[4,1])$$

comme le montre le graphe suivant



Le polynôme de spin est ensuite donné par

$$G^{(k)}_{\lambda\mu}(q) = q^{n(k-1)/2} F^{(k)}_{\alpha\mu}(q^{\frac{1}{2}}) \ .$$

C'est un vrai polynôme (sans exposants fractionnaires) si toutes les parts de λ sont multiples de k (c'est la situation la plus intéressante).

References

[1] J. Désarménien, B. Leclerc & J.-Y. Thibon, *Hall-Littlewood functions and Kostka-Foulkes polynomials in representation theory*, Rapport IGM 94-13, à paraître aux Actes du Séminaire Lotharingien de Combinatoire.

[2] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, V. Sunderam, *PVM: Parallel Virtual Machine, A users guide and tutorial for networked parallel computing*, MIT Press, 1994.

[3] D. Knuth, *The art of computer programming*, vol. 3, Addison-Wesley, 1981.

[4] D. Stanton & D. White, *A Schensted algorithm for rim-hook tableaux*, J. Comb. Theory A **40** (1985), 211-247.