Université de Paris 7 **(F.Han[1])**

TP03 Anneaux principaux

agreg 2010

Exercice I:

1) Quelle est la bonne instruction pour calculer le pgcd de 2 entiers, 3 entiers, et celle de 2 polynômes ?

2) Les majuscules et les minuscules en mode maple : essayer : `gcd(2*X,4*X) ; Gcd(2*X,4*X) ; evala(Gcd(2*X,4*X));`
`gcdex(2*x^3,9*x^2+x,x,'a','b');a;b;` (où a et b sont des variables libres)

3) L'intérêt de ne pas évaluer tout de suite : `gcd(x+2,x) mod 2 ;` et `Gcd(x+2,x) mod 2 ;`. Etudier la documentation sur le calcul modulaire.

4) a) Comment trouver le quotient et le reste d'une division d'entiers ? Quel est le pgcd d'un entier avec 0 ?

b) Pour les entiers : Quelle instruction permet de récupérer un couples de bezout ?

5) a) Comment détruire une ligne ou une colonne d'une matrice ?

b) Quelle instruction permet de trouver une matrice diagonale $GL(\mathbb{Z})$ équivalente à un élément de $\mathcal{M}_n(\mathbb{Z})$ donné ?

Exercice II: Matrices $GL(\mathbb{Z})$-équivalentes

[2] (Cf par exemple leçon "opérations lignes/colonnes") L'algorithme permettant d'obtenir les facteurs invariants : On le fait sur $\mathbb{Z}$ ou dans le cadre Euclidien, car on n'a pas d'algorithme pour obtenir les nombres de Bezout dans le cadre principal.

Cet algorithme est déjà programmé dans xcas sur $\mathbb{Z}$. (`ismith`) On pourrait aussi le faire dans $k[x]$, mais nous nous limiterons ici à $\mathbb{Z}$. Nous allons procéder en quelques étapes. 1a) Trouver un minimum non nul (au sens du stathme) de la matrice. 1c) par opérations lignes/colonnes on fait diminuer les autres coefficients de sa ligne et de sa colonne. 2) On recommence 1a) et 1c) tant que c'est possible. lorsque l'on a fini, le minimum n'a que des 0 sur sa ligne et sur sa colonne. On garde cette valeur, on raye la ligne et la colonne, et l'on recommence avec la matrice plus petite. On obtient alors une matrice diagonale équivalente (dans l'anneau) à la matrice de départ, ce qui est déjà bien. 3) On peut passer de cette forme aux diviseurs élémentaires par lemme chinois, ou en continuant l'algorithme de maniere un peu modifiée.

1) a) Faire une procédure `minval` qui donne le couple (i,j) tel que $A_{i,j}$ soit un coefficient de la matrice A ayant la plus petite valeur absolue non nulle. (Elle pourra éventuellement retourner $(0,0)$ si A est la matrice nulle)

b) Prendre[3] un exemple à coefficients entiers noté A. Repérer son plus petit coefficient non nul : $A_{i,j}$ en valeur absolue, et créer une matrice $U \in GL(\mathbb{Z})$ telle que $(A.U)_{i,l}$ soit le reste de la division de $A_{i,l}$ par $A_{i,j}$ pour $l \neq j$, et $(A.U)_{i,j} = A_{i,j}$.

c) Faire une procédure `trans(A,i,j)` qui pour une matrice A telle que $A_{i,j} \neq 0$ donne une matrice A', $GL(\mathbb{Z})$-équivalente à A, telle que $\forall l \in \{1 \dots n\} - \{j\}$, $|A'_{i,l}| < |A_{i,j}|$ et $\forall l \in \{1 \dots n\} - \{i\}$ $|A'_{l,j}| < |A_{i,j}|$.

2) a) Essayer[4] de programmer la partie qui trouve une matrice diagonale $GL(\mathbb{Z})$-équivalente à A. On appellera cette procédure `Zequiv`. (On remarquera qu'une matrice dont les coefficients non nuls sont les seuls coefficients non nuls de leur ligne et de leur colonne est équivalente via des permutations à une matrice diagonale. Dans notre procédure on ne cherchera pas à trouver cette permutation.)

b) Obtient t'on forcément les diviseurs élémentaires ?

3) a) Dans le cas où A est une matrice du type $\begin{pmatrix} d_1 & \dots & d_n \\ 0 & \ddots & 0 \\ 0 & 0 & d_n \end{pmatrix}$ telle que d_1 soit un élément de valeur absolue minimale parmi les coefficients non nuls de A, quelle est la valeur du minimum de la valeur absolue des coefficients non nuls de `ZequivC(A)` ? (Où `ZequivC` est une version de `Zequiv` qui ne travaille que sur les colonnes. ie, on ne touche qu'à la première ligne)

b) En déduire que l'on peut obtenir les diviseurs élémentaires d'une matrice diagonale de taille n en appliquant au plus n fois `ZequivC(C)` à des matrices bien choisies. Créer une procédure `elem` qui le fasse.

[1] http ://www.math.jussieu.fr/∼han/agreg

[2] $A \sim B \iff \exists P, Q \in GL(\mathbb{Z}),\ P^{-1}.A.Q = B$

[3] On s'entraine à la main sur un exemple, pour mieux faire la procédure de la question suivante

[4] Remarquer que la décroissance stricte oblige l'algorithme à se terminer, ce qui donne une preuve du résultat suivant : Tout $\mathbb{Z}$-module de type fini se décompose en produit de module monogènes. Du même genre : tout groupe abélien fini est isomorphe à un produit de groupe cyclique.