Cours d'Algèbre et Calcul Formel

Ecrit par Marion Candau

Enseignant : M.Karim Belabas

Master 1 Cryptologie et Sécurité Informatique
Université Bordeaux 1

2009 - 2010

Table des matières

Chapitre 1

Opérations et structures fondamentales

1.1 $\mathbb{Z}$ et R[X], R anneau commutatif unitaire

Représentation des données :

- Dans $\mathbb{Z}$, on fixe une base $\beta > 1$, en pratique 2,10 ou 2^{64}. Tout entier $A \geqslant 1$ s'écrit de façon unique $A = \sum_{i=0}^{n} a_i \beta^i$ avec $a_i \in \{0, 1, \ldots, \beta - 1\}$ et $a_n \neq 0$. On a $n = \lfloor log_\beta A \rfloor + 1$.
- Dans R[X], tout $A \neq 0$ s'écrit $A = \sum_{i=0}^{n} a_i X^i$, avec $a_i \in R, a_n \neq 0$.

On traite 0 séparément. La taille de A est l'entier n+1
→ complexité mot-machine sur $\mathbb{Z}$ (binaire : $\beta = 2$)
→ complexité algébrique sur R[X]

1.2 Addition/Soustraction

Définition
On appelle complexité algébrique d'une opération le nombre d'opérations $(+, -, \times, /)$ dans l'anneau de base qu'elle utilise.
On appelle complexité binaire d'une opération le nombre d'opérations $(+, -, \times, /)$ sur les mots-machines (chiffres) qu'elle utilise.

Algorithme 1 Addition dans R[X]

Entrées: $A = \sum_{i=0}^{n_A} a_i X^i$, $B = \sum_{i=0}^{n_B} b_i X^i$

Sorties: $C = A + B \in R[X]$

Pour $i = 0$ à $max(n_A, n_B)$ **Faire**

poser $c_i \leftarrow a_i + b_i$

Fin pour

Retourner $C = \sum_{i=0}^{max(n_A,n_B)} c_i X^i$, normalisé

Coût : $1 + max(n_A, n_B)$ opérations dans R.

Algorithme 2 Addition dans $\mathbb{Z}$

Entrées: $A = \sum_{i=0}^{n_A} a_i \beta^i$, $B = \sum_{i=0}^{n_B} b_i \beta^i$

Sorties: $C = A + B$

$r \leftarrow 0$

Pour $i = 0$ à $max(n_A, n_B)$ **Faire**

écrire $a_i + b_i + r = c_i + r'\beta$, $0 \leqslant c_i < \beta$, $r' = 0$ ou 1

$r \leftarrow r'$

Fin pour

Retourner $c = \sum_{i=0}^{max(n_A,n_B)} c_i \beta^i + \begin{cases} \beta^{max(n_A,n_B)+1} & \text{si } r = 1 \\ 0 & \text{si } r = 0 \end{cases}$

Coût : $max(n_A, n_B) + 1$ opérations sur des mots-machines.

Corollaire

L'addition dans $\mathbb{Z}$ ou R[X] s'effectue en complexité binaire ou algébrique, linéaire en la taille des opérandes.

Remarque

1. En pratique l'algorithme asymptotiquement meilleur ne l'est pas en "petites tailles".
2. En pratique 2^{80} opérations $= +\infty$

1.3 Multiplications

Algorithme 3 Multiplication dans R[X]

Entrées: $A = \sum_{i=0}^{n_A} a_i X^i$, $B = \sum_{i=0}^{n_B} b_i X^i$ non nuls

Sorties: $C = A \times B = \sum_{k=0}^{n_C \leqslant n_A+n_B} c_k X^k$ et $c_k = \sum_{i+j=k} a_i b_j$

Pour $k = 0$ à $n_A + n_B$ **Faire**
  $c \leftarrow 0$;
  Pour $i = 0$ à k **Faire**
    $c \leftarrow c + a_i \times b_{k-i}$
    $c_k \leftarrow c$
  Fin pour
Fin pour

Retourner $C = \sum_{k=0}^{n_A+n_B} c_k X^k$, normalisé

Complexité algébrique

$$\sum_{k=0}^{n_A+n_B} \sum_{i=0}^{k} 2 = O\left[(n_A + n_B)^2\right]$$

Remarque

$$\sum_{k=0}^{N} k^\alpha = O(N^{\alpha+1})$$

$$\#\{a_i b_j\} = (n_A + 1)(n_B + 1)$$

Théorème
La complexité algébrique de cet algorithme est $\leqslant 2 \times (n_A + 1)(n_B + 1) = O\left((n_A + 1)(n_B + 1)\right)$.

Définition : Landau $f = O(g)$ au voisinage de $+\infty$ si $\exists x_0$ tel que $\forall x > x_0$ on a :

$$|f(x)| \leqslant c(x_0)|g(x)|$$

Définition
$f \ll g$ si pour tout x on a : $|f(x)| \leqslant c|g(x)|$.

Algorithme 4 Sous Algorithme de multiplication dans $\mathbb{Z}$

Entrées: $A = \sum_{i=0}^{n_A} a_i\beta^i$, $b_i \in \{1, \ldots, \beta - 1\}$
Sorties: $A \times b = C = \sum c_i\beta^i$
  Pour $i = 0$ à n_A **Faire**
    écrire $a_ib + r = q\beta + s$, $0 \leqslant s < \beta$, $0 \leqslant q < \beta$
    $r \leftarrow q$;
    $c_i \leftarrow s$;
  Fin pour
  Retourner $r\beta^{n+1} + \sum_{i=0}^{n} c_i\beta^i$, normalisé

Algorithme 5 Multiplication dans $\mathbb{Z}$

Entrées: $A = \sum_{i=0}^{n_A} a_i\beta^i$, $B = \sum_{i=0}^{n_B} b_i\beta^i$ non nuls
Sorties: $A \times B$
  Pour $i = 0$ à n_B **Faire**
    $d_i \leftarrow b_i \times A$
  Fin pour
  Retourner $c = \sum_{i=0}^{n_B} d_i\beta^i$
  Coût $\ll \sum_j n_A + 2 = (n_A + 2)(n_B + 1) = o(\text{taille(A)}\times\text{taille(B)})$

Corollaire
La multiplication dans $\mathbb{Z}$ ou R[X] s'effectue en complexité binaire ou algébrique en la taille de A $\times$ la taille de B.

1.4 Division Euclidienne

Dans R[X], $\exists A, B$ tel que $cd(B) \in R^*(\Rightarrow B \neq 0)$ alors $A = Bq + r$ avec $deg(r) < deg(B)$, $q, r \in R[X]$ uniques.

Complexité algébrique
Supposons $n_A \geqslant n_B$. On a $O((n_A - n_B + 1)(n_B + 1))$ opérations dans R $= O(\text{taille(A)}\times\text{taille(B)})$
Si $deg(A) \leqslant 2n$ et $deg(B) \leqslant 2n$ on a une complexité en $O(n^2)$.

Algorithme 6 Algorithme de division euclidienne

Entrées: $A = \sum_{i=0}^{n_A} a_i X^i$, $B = \sum_{i=0}^{n_B} b_i X^i$

Sorties: $A/B = q$ et $A \mod B = r$ tel que $q = \sum_{i=0}^{n_A - n_B} q_i X^i$

$r \leftarrow A$; $q \leftarrow 0$;
Pour $i = n_A - n_B$ à 0 **Faire**
  Si $deg(r) = n_B + i$ **Alors**
    $q_i \leftarrow \frac{cd(r)}{cd(B)}$;
    $r \leftarrow r - q_i X^i - B$
  Sinon
    $q_i \leftarrow 0$
  Fin si
Fin pour
Retourner $q = \sum_{i=0}^{n_A - n_B} q_i X^i$ si $n_A \geqslant n_B$, $q = 0$ sinon et r, normalisés

Division euclidienne dans $\mathbb{Z}$

Soient $A \in \mathbb{Z}, B \in \mathbb{Z}, B \neq 0$. Il existe q, r uniques tels que $A = Bq + r$ avec $0 \leqslant r < |B|$ ou $-\frac{|B|}{2} < r \leqslant \frac{|B|}{2}$

Proposition

O_β(taille(q)×taille(B)) opérations sur les mots.

1.5 Algorithmes sous-quadratiques

Théorème

Soit $M_{\mathbb{Z}}(n)$ le nombre maximal d'opérations sur les mots pour multiplier 2 opérandes de taille $\leqslant n$ dans $\mathbb{Z}$ et $M_{R[X]}(n)$ le nombre maximal d'opérations sur les mots pour multiplier 2 opérandes de taille $\leqslant n$ dans $R[X]$.

1. Karatsuba 1962
$$M_{\mathbb{Z}}(n) = O(n^{\log_2 3})$$
2. Schonhage-Staβen 1971
$$M_{\mathbb{Z}}(n) = O(n \log n \log\log n) = \tilde{O}(n)$$
Remarque : $\tilde{O}(f) = f(\log f)^{O(1)}$

3. Frer 2005
$$M_{\mathbb{Z}}(n) = O(n \log n 2^{\log_* n})$$
Remarque : $\log_*(n) = \min_{k \geqslant 0}\{k, 2 \text{ puiss } 2 \text{ puiss } 2 \ldots \text{ puiss } 2 \text{ k fois } > n\}$
4. Cantor-Kaltofen 1991
$$M_{R[X]}(n) = O(n \log n \log\log n) = \tilde{O}(n)$$

Théorème
Soit $D_{\mathbb{Z},R[X]}(n)$ la complexité de la division euclidienne sur 2 opérandes de taille $\leqslant n$.
$$D_\bullet(n) = O(M_\bullet(n))$$

Théorème
Soit $f : \mathbb{R}^+ \longrightarrow \mathbb{R}^+$ telle que $b > 1, a > 0, c \in \mathbb{R}$. On a :
$$f(a) \leqslant af\left(\frac{x}{b}\right) + cx$$
avec $f(x) = 1$ si $x < 1$.
Alors
$$f(x) \ll \begin{cases} O(x^{\log_b(a)}) & \text{si } a > b \\ O(x \log x) & \text{si } a = b \\ O(x) & \text{si } a < b \end{cases}$$
On écrit $(a_1X + a_0)(b_1X + b_0) = a_1b_1X^2 + (a_0b_1 + a_1b_0)X + a_0b_0$.
Or $(a_0b_1 + a_1b_0) = (a_0 + a_1)(b_0 + b_1) - a_1b_1 - a_0b_0$.
Avec $A(x) = a_1(x)x^n + a_0(x)$ et $B(x) = b_1(x)x^n + b_0(x)$ et $da_i, b_i < n$, on a $A \times B$ qui se calcule en 3 multiplications et $O(1)$ additions de polynomes de degré $< n$.
Si $M(s)$ est la complexité algébrique de la multiplication de 2 polynomes de degré $< s$, on peut supposer $s = 2^k$, alors
$$M(s) \leqslant 3M\left(\frac{s}{2}\right) + O(s) \Rightarrow M(s) = O(s^{\log_2 3})$$

Remarque
Si s n'est pas une puissance de deux :
1. $M(s) \leqslant 3M\left(\left\lceil \frac{s}{2} \right\rceil\right) + O(s)$.
2. On remplace s par $2^{\lceil \log_2 s \rceil} \geqslant s$ et on applique le lemme d'où :
$$M(s) = O(2^{\lceil \log_2 s \rceil log_2 3}) \ll 2^{(\log_2 s + 1)\log_2 3} = 3 \times s^{\log_2 3}$$

Remarque

Soient $\alpha(x) = a_1(x) + a_0$ et $\beta(x) = b_1(x) + b_0$, on cherche $\gamma(x) = \alpha\beta = c_2X^2 + c_1X + c_0$. On fixe x_1, x_2, x_3 distincts dans un corps de base.
On calcule $\gamma(x_i) = \alpha(x_i)\beta(x_i)$, $i = 1, 2, 3$ + Lagrange pour retrouver c_0, c_1, c_2.
Cette méthode se généralise avec r morceaux et on a une complexité de $O(s^{1+\epsilon(r)})$, $\lim_{r\to\infty} \epsilon(r) = 0$.

Cas particulier : FFT

Hypothse

On travaille sur un corps $K[X]$ qui contient ω une racine de l'unité d'ordre exact n.

Définition : DFT

Soit $T \in K[X]$, on définit

$$\mathcal{F} : K[X]/X^n - 1 \longrightarrow K^n$$

$$T \longmapsto (T(\omega^0), T(\omega^1), \ldots, T(\omega^{n-1})) = \mathcal{F}(T, \omega)$$

Lemme

$\mathcal{F}$ ralise un isomorphisme de K-algèbre

$$(K[X]/X^n - 1), \times) \simeq (K^n, \text{ produit composante par composante})$$

Corollaire

$$TU = \mathcal{F}^{-1}(\mathcal{F}(T) \otimes \mathcal{F}(U))$$

Ceci permet de calculer TU dans $K[X]/X^n - 1$, TU dans $K[X]$ si $d(T) < \frac{n}{2}$ et $d(U) < \frac{n}{2}$.

Lemme

Si $T \in K[X], \mathcal{F}(T, \omega) = (u_1, \ldots, u_{n-1})$, et $U(X) = \sum_{i=1}^{n-1} u_i X_i$. Alors $\mathcal{F}(U, \omega^{-1}) = nT$.

Théorème

FFT s'effectue en $O(n \log n)$ opérations dans K.

Algorithme 7 Algorithme FFT

Entrées: $n = 2^k$, $T \in K[X]$, $dT < n$, ω d'ordre n dans K
Sorties: $\mathcal{F}(T, \omega)$
Si $n = 1$ retourner T.
Ecrire $T(X) = T_0(X^2) + XT_1(X^2)$
Calculer $\mathcal{F}(T_0, \omega^2) = (a_0, \ldots, a_{\frac{n}{2}-1})$ et $\mathcal{F}(T_1, \omega^2) = (b_0, \ldots, b_{\frac{n}{2}-1})$
Retourner $(a_i + \omega^i b_i)_{i<n}$ où $(a_i), (b_i)$ sont étendus par périodicité de période $\frac{n}{2}$.

Remarque

1. si $\omega \notin K$, il faut le rajouter (Schonhage-Strassen)
2. On peut définir un algorithme FFT_k en coupant les entrées en k morceaux.

Divisions sous -quadratiques

Soient $a, b \in R[X]$, b unitaire, $da = n$, $db = m \leqslant n$.
Si $a = bq + r$ avec $dr \leqslant m - 1$ et $dq \leqslant n - m$, alors

$$X^n a\left(\frac{1}{X}\right) = X^m b\left(\frac{1}{X}\right) X^{n-m} q\left(\frac{1}{X}\right) + X^n r\left(\frac{1}{X}\right)$$

$$A = BQ + X^{n-m+1}(*)$$

On calcule $Q \equiv AB^{-1}$ dans $R[X]/X^{n-m+1} \to q \to r = a - bq$.

Théorème
Soit $B, C_0, C_1, \ldots \in R[X]$ tels que $B(0) = 1$, $C_0 = 1$ et $C_{i+1} \equiv 2C_i - BC_i^2 \mod X^{2^{i+1}}$.
Alors $BC_i \equiv 1 \mod X^{2^i}$

Définition
Soit R un anneau commutatif. On définit $R[[X]]$ comme $R^{\mathbb{N}}$ muni de la structure d'anneau suivante :

$$\sum a_i X^i + \sum b_i X^i = \sum (a_i + b_i) X^i$$

$$\sum a_i X^i \times \sum b_i X^i = \sum c_k X^k \text{ avec } c_k = \sum_{i+j=k} a_i b_j$$

$R[X]$ est un sous-anneau de $R[[X]]$
Si $a_0 \in R^* \Leftrightarrow \sum_{i \geqslant 0} a_i X^i \in R[[X]]^*$.

Algorithme 8 Algorithme inversion

Entrées: $B \in R[X]$ tel que $B(0) = 1$, $l \in \mathbb{N}$ (précision)
Sorties: $C \in R[X]$ tel que $BC \equiv 1 \mod X^l$

1: $r \leftarrow \lceil \log_2 l \rceil$ $c_0 \leftarrow 1$
2: **Pour** $i = 1$ à r **Faire**
3: $C_i \leftarrow (2C_{i-1} - BC_{i-1}^2)$ rem X^{2^i}
4: **Fin pour**
5: **Retourner** $c = c_r(\text{rem } X^l)$

Théorème

On suppose que $M_{R[X]} = M$ vérifie $M(2n) \geqslant 2M(n)$ et M croissante. Alors l'algorithme d'inversion utilise $O(M(2^r)) = O(M(2l))$ opérations dans R.

Algorithme 9 Algorithme division euclidienne

Entrées: $a, b \in R[X]$, b unitaire
Sorties: $q, r \in R[X]$ tels que $a = bq + r$, $dr \leqslant db$

1: **Si** $da < db$ **Alors**
2: **Retourner** $(0, a)$
3: **Fin si**
4: $l \leftarrow da - db + 1 \geqslant 1$
5: Calculer $C \in R[X]$ tel que $BC \equiv 1 \mod X^l$ où $B(X) = X^{dB} b\left(\frac{1}{X}\right)$
6: Calculer $Q \leftarrow AC \mod X^l$ où $A(X) X^{dA} a\left(\frac{1}{X}\right)$
7: **Retourner** $q = X^{dAdB} Q\left(\frac{1}{X}\right)$ et $r = a - bq$.

1.6 Structures fondamentales

1.6.1 Fractions $\mathbb{Q}, K[X]$

Soit R un anneau intègre, on définit Frac R= $R \times (R \setminus \{0\}) / \sim$ où $(a, b) \sim (c, d) \Leftrightarrow ad = bc$.

Notation

$\frac{a}{b} = (a, b)$dans Frac R. On le munit d'une structure d'anneau :

$$\frac{a}{b} \times \frac{c}{d} = \frac{ac}{bd}$$

$$\frac{a}{b} + \frac{c}{d} = \frac{ad + bc}{bd}$$

Théorème : Frac R est un corps.

Normalisation

On peut imposer $pgcd(a, b) = 1$.

$$(a, b) = \left(\frac{a}{pgcd(a, b)}, \frac{b}{pgcd(a, b)} \right) \text{ dans Frac R}$$

1.6.2 Quotients $\mathbb{Z}/n\mathbb{Z}, K[X]/(T)$

Soit R un anneau, I un idéal de R, R/I l'anneau quotient où $a \sim b \Leftrightarrow a - b \in I$.

Notation

$$a \equiv b \mod I$$

Cas particulier

On se restreint au cas où R est euclidien $(\mathbb{Z}, K[X])$.

* Tout idéal I est principal, $I = (x) = x.R$. Notation : $a \equiv b \mod x$.
* Toute classe $a + (x) \in R/(x)$ contient un élément canonique.

Définition

1. $a + N\mathbb{Z} \in \mathbb{Z}/N\mathbb{Z}$ est sous forme normale si et seulement si $0 \leqslant a < |N|$.
2. $a + TK[X] \in K[X]/(T)$ est sous forme normale si et seulement si $da < dT$.

Remarque

Pour mettre une représentation quelconque $a + xR$ sous forme normale il suffit de remplacer a par $rem(a, x)$.
$a + (x)$ et $a' + (x)$ sous formes normales sont égales si et seulement si $a = a'$ dans R.

Corollaire

+, - dans $R/(x)$ s'effectuent avec complexité $O(\text{taille}(x))$.
$\times$ dans $R/(x)$ s'effectue avec complexité $O(M_R(\text{taille}(x)))$.

1.7 Complétions $\mathbb{R}, K[[T]]$

$$K[[T]] = \text{Frac } (K[[T]]) = \left\{ \sum_{n \geqslant n_0} a_n T^n : a_n \in K, n_0 \in \mathbb{Z} \right\}$$

1.7.1 $\mathbb{R}$

Définition

Soit $l \in \mathbb{N}$ fixé (précision), e_{min}, $e_{max} \in \mathbb{Z}$ fixés. L'ensemble des nombres flottants en précision l est :

$$\mathcal{F} = \mathcal{F}_{l,e_{min},e_{max},\beta} = \{0\} \cup \{\pm\beta^e.m : e \in [e_{min}, e_{max}], \beta^{l-1} \leqslant m \leqslant \beta^l\}$$

On choisit une fonction d'arrondi $A : \mathbb{R} \to \mathcal{F} \cup \{\text{erreur}\}$

$$A^-(x) = \begin{cases} \max\{f \in \mathcal{F}, f \leqslant x\} & \text{par défaut} \\ \text{erreur} & \text{si l'ensemble est vide} \end{cases}$$

$$A^+(x) = \begin{cases} \min\{f \in \mathcal{F}, f \geqslant x\} & \text{par défaut} \\ \text{erreur} & \text{si l'ensemble est vide} \end{cases}$$

Définition

Soient $x, y \in \mathcal{F}$

$$x \oplus y = A(x + y)$$
$$x \ominus y = A(x - y)$$
$$x \otimes y = A(x \times y)$$
$$x \oslash y = A(x/y)$$

définit des opérations de $\mathcal{F} \times \mathcal{F} \to \mathcal{F} \cup \{\text{erreur}\}$

Théorème

$x + y$ ou $x - y$ se calcule en $O(l)$ opérations algébriques dans $\mathbb{Z}$.
$x \times y$ ou x/y se calcule en $O(M(l))$ opérations algébriques dans $\mathbb{Z}$.

1.7.2 $K[[T]]$

Définition

L'ensemble des "séries flottantes" est :

$$\mathcal{F} = \{0\} \cup \{T^e.m, e \in [e_{min}, e_{max}], m \in K[T], dm \leqslant l-1\}$$

et la fonction $K[[T]] \to \mathcal{F} \cup \{\text{erreur}\}$

$$T^e \sum_{i \geqslant 0} m_i T^i \mapsto T^e \sum_{0 \leqslant i < l} m_i T^i$$

si $e \in [e_{min}, e_{max}]$.

1.8 Nombres algébriques

Définition

Soit $K \subset$ un corps (commutatif).
$\alpha \in L$ est algébrique sur K s'il existe $P \in K[X], P \neq 0, P(\alpha) = 0$.

Proposition

Si α est algébrique alors il existe $P \in K[X]$, P unitaire, unique tel que

$$P(\alpha) = 0$$

$$Q(\alpha) = 0 \Rightarrow P|Q$$

P est appelé polynome minimal de α.

Corollaire

$$K[\alpha] \simeq K[X]/(P_{min})$$
$$\alpha \mapsto X + (P_{min})$$
$$A(\alpha) \leftarrow A(X)$$

Cas particulier : $K = \mathbb{F}_p$

Si $P \in \mathbb{F}_p[X]$ est irréductible de degré n alors $\mathbb{F}_p[X]/(P) \simeq \mathbb{F}_p^n$.

Théorème

$$\#\{P \text{ unitaire } \in \mathbb{F}_p[X], P \text{ irréductible de degré n } \} = \frac{1}{n}\sum_{d|n}\mu(d)q^{\frac{n}{d}} > 0$$

où $\mu(d) = \begin{cases} -1 & \text{si } d = p_1 \dots p_k, p_i \text{ premiers différents} \\ 0 & sinon \end{cases}$

Corollaire

- Les éléments de $\mathbb{F}_q$ se codent avec $O(\log q)$ chiffres
- $+, -$ ont pour complexité binaire $O(\log q)$
- $\times$ a pour complexité binaire $\tilde{O}(\log q)$.

Chapitre 2

Quelques algorithmes arithmétiques fondamentaux

2.1 Euclide

Algorithme 10 Algorithme d'Euclide

Entrées: $a, b \in \mathbb{Z}$ ou $K[X]$
Sorties: $pgcd(a, b)$

1: $r_0 \leftarrow a$;
2: $r_1 \leftarrow b$;
3: $i \leftarrow 1$;
4: **Tant que** $r_i \neq 0$ **Faire**
5: $\quad r_{i+1} \leftarrow rem(r_{i-1}, r_i)$;
6: $\quad i \leftarrow i + 1$
7: **Fin tant que**
8: **Retourner** r_{i-1}

Remarque

$$(a, b) \leftarrow (b, a \mod b)$$

$$\begin{pmatrix} a \\ b \end{pmatrix} \leftarrow \begin{pmatrix} 0 & 1 \\ 1 & -q \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix}$$

Théorème

Euclide utilise $O(n^2)$ opérations élémentaires dans $\mathbb{Z}$ (ou $K[X]$) si a, b sont de taille $\leqslant n$.

Algorithme 11 Algorithme d'Euclide tendu

Entrées: $a, b \in \mathbb{Z}$ ou $K[X]$

Sorties: $pgcd(a,b), u, v$ tels que $au + bv = pgcd(a,b)$

1: $r_0 \leftarrow a$;
2: $r_1 \leftarrow b$;
3: $i \leftarrow 1$;
4: $U_1 \leftarrow Id_1$
5: **Tant que** $r_i \neq 0$ **Faire**
6: $\quad r_{i+1} \leftarrow rem(r_{i-1}, r_i)$;
7: $\quad U_{i+1} \leftarrow \begin{pmatrix} 0 & 1 \\ 1 & -q_{i+1} \end{pmatrix} U_i$
8: $\quad i \leftarrow i + 1$
9: **Fin tant que**
10: **Retourner** $r_{i-1} = pgcd(a,b)$ et la première ligne de U_{i-1}, $ua + vb = pgcd(a,b)$

Théorème

Cet algorithme utilise $O(n^2)$ opérations élémentaires si taille(a) et taille(b) $\leqslant n$.

Théorème

Il existe un algorithme calculant une relation de Bezout en $O(M(n)\log n) = \tilde{O}(n)$ opérations.

Application 1

Soient $R = \mathbb{Z}$ ou $K[X]$, $a, b \in R$ tels que $PGCD(a,b) = 1$. Alors $\exists u, v \in R$ tels que $au + bv = 1$, et $\bar{u} = \bar{a}^{-1}$ dans $R/(b)$.

Application 2

Soient $a, b \in R$, $PGCD(a,b) = 1$ et $au + bv = 1$. Alors :

$$R/(ab) \simeq R/(a) \times R/(b)$$
$$x \mapsto (x, x)$$
$$au\beta + bv\alpha \leftarrow (\alpha, \beta)$$

Application 2'

Soient $a_1, \ldots, a_n \in R$, $PGCD(a_i, a_j) = 1$, $\forall i \neq j$. Alors :

$$R/\left(\prod_{i=1}^{n} a_i\right) \simeq \prod_{i=1}^{n} R/(a_i)$$

$$x \mapsto (x, \ldots, x)$$

$$\sum_{i=1}^{n} \alpha_i u_i A_i \leftarrow (\alpha_1, \ldots, \alpha_n)$$

avec $A_i = \prod_{j \neq i} a_j$ et $PGCD(a_i, A_i) = 1 = u_i A_i + v_i a_i = 1$.

Remarque

Soient $R = K[X]$, $a_i = X - \alpha_i$, avec $\alpha_i \in K$, et $\alpha_i \neq \alpha_j$ pour $i \neq j$. Alors :

$$K[X]/\prod_{i=1}^{n}(X - \alpha_i) \simeq \prod_{i=1}^{n} K[X]/(X - \alpha_i) \simeq K^n$$

$$P(X) \mapsto (P(X), \ldots, P(X)) \simeq (P(\alpha_1), \ldots, P(\alpha_n))$$

Application 3 : Interpolation d'Hermite

Soient $\alpha_1, \ldots, \alpha_k$ deux à deux distincts dans K. On veut $P \in K[X]$ tel que $(*)$:

$$P(\alpha_1), P'(\alpha_1), \ldots P^{(l_1)}(\alpha_1)$$
$$P(\alpha_2), P'(\alpha_2), \ldots P^{(l_2)}(\alpha_2)$$
$$\vdots$$
$$P(\alpha_n), P'(\alpha_n), \ldots P^{(l_n)}(\alpha_n)$$

prennent des valeurs fixées à l'avance.

$$P(\alpha) = \beta^{(0)}$$
$$P'(\alpha) = \beta^{(1)}$$
$$\vdots$$
$$P^{(l)}(\alpha) = \beta^{(l)}$$
$$\iff P(\alpha) = \beta^{(0)} + \beta^{(1)}(X - \alpha) + \ldots + \beta^{(l)}\frac{(X - \alpha)^l}{l!} + \cdots + (X - \alpha)^k$$

Or $+ \cdots + (X - \alpha)^k$ est divisible par $(X - \alpha)^{l+1}$ donc $P(\alpha)$ est connu modulo $(X - \alpha)^{l+1}$.

$$(*) \Leftrightarrow P = Q_1 \mod (X - \alpha)^{l_1+1}$$
$$\vdots$$
$$\iff P = Q_n \mod (X - \alpha)^{l_n+1}$$

où les Q_i sont connus. C'est donc un problème chinois dans $K[X]$.

Application 4 : algorithme modulaire

Soit $A = (a_{ij}) \in M_n(\mathbb{Z})$, quel est $\det(A)$?

Soit $\varphi : \mathbb{Z} \longrightarrow R$ morphisme d'anneau. Alors $\varphi(\det(A)) = \det(\varphi(A))$.

1. Soit $\varphi : \mathbb{Z} \longrightarrow \mathbb{F}_p$ projection canonique. On connait alors $\det(A) \mod p$ après un pivot de Gauss sur $A \mod p$ dans $M_n(\mathbb{F}_p)$
2. On obtient donc $\det(A) \mod (p_1, \ldots, p_k)$ si les p_i sont des premiers différents.
3. Si $|\det A| < B$ et $N \geqslant 2B$, alors connaitre $\det(A) \mod N$ détermine $\det(A)$ dans $\mathbb{Z}$.
$$|\det(A)| \leqslant n!(\|A\|_\infty)^n$$

On veut $\displaystyle\sum_{i=1}^{k} \log p_i > \log(2B) = n\log(2\|A\|_\infty) + \log n!$.

Parenthèse sur les nombres premiers

Théorème des nombres premiers

$$\theta(x) = \sum_{p \leqslant x, p \text{ premier}} ln(p) \underset{x\to\infty}{\sim} x$$

$$\Pi(x) = \#\{p \leqslant x, p \text{ premier}\} \sim \frac{x}{ln(x)}$$

Remarque

Tout est effectif : $\theta(x) > 0,98x$ pour $x > 7481$.

D'où $\{p_i\} = \{p \leqslant x, \text{ premier}\}$ et :

$$\sum_{p \leqslant x} \log p = \theta(x) > 0,98x$$

si $0,98x > \log(2B) \Rightarrow$ OK.

Remarque

$$\sum_{\frac{x}{2} \leqslant p < x} ln(p) = x - \frac{x}{2} \sim \frac{x}{2}$$

d'où :

$$\#\left\{\frac{x}{2} \leqslant p < x, p \text{ premier} \sim \frac{x}{2ln(x)}\right\}$$

2.2 Crible d'Eratosthène

Algorithme 12 Crible d'Eratosthène

Entrées: $B \in \mathbb{N}$
Sorties: $\{p \leqslant B, p \text{ premier}\}$
1: Initialiser $T[2] = \ldots = T[B] = true$
2: **Pour** $n = 2$ à $\sqrt{B}$ (tel que $T[n] = true$) **Faire**
3: **Pour** $k = 2$ à $\lfloor \frac{B}{n} \rfloor$ **Faire**
4: $T[kn] \leftarrow false$;
5: **Fin pour**
6: **Fin pour**
7: **Retourner** $\{i \leqslant B, T[i] = true\}$

Améliorations

- Si $T[n] = false$ alors n n'est pas premier et ses multiples sont déjà barrés.
- Si $k < n$ alors $T[kn]$ est déjà barré.
- Et si on veut les nombres premiers entre A et $A + B$?
  translation faire : $T[i]$ associé $A + i$.
- Si B est grand ?
  1. $\{p \leqslant \sqrt{A+B}\}$
  2. Tableaux associés $\left[A, A + \frac{B}{N}\right], \left[A + \frac{B}{N}, A + \frac{2B}{N}\right]$
- On fixe $N = 2 \times 3 \times 5$, on utilise un crible pour chacune des $\varphi(N)$ classes (mod N) dans lesquelles se trouvent les nombres premiers.
  gain : N en mémoire et $\frac{N}{\varphi(N)}$ en temps.

Le nombre d'accès mémoire de cet algorithme est alors :

$$B + \sum_{n \leqslant \sqrt{B}} \left\lfloor \frac{B}{N} \right\rfloor + B = O(B) + O(\sqrt{B}) + B \sum_{n \leqslant \sqrt{B}} \frac{1}{n}$$

$$= \begin{cases} O(B) + O(\sqrt{B}) + Bln(\sqrt{B}) & \text{sans n premier} \\ O(B) + O(\sqrt{B}) + B(ln(ln(\sqrt{B})) & \text{avec n premier} \end{cases} \sim Bln(ln(B))$$

Le coût par nombre premier est $\sim ln(B)ln(ln(B))$.

2.3 Exponentiation binaire

Soit G un ensemble muni d'une loi associative $\times$. Soit $g \in G$, on veut calculer g^n avec $n \geqslant 1$ un entier.

Ides

$$\underbrace{g \times g \times g \times \ldots \times g}_{\text{n-1 fois}} = g^n$$

$$\left(\left(\left(\left(g^2\right)^2\right)^2\right)^{\cdots}\right)^2 = g^{2^k}$$

Donc pour $n = 2^k$ une puissance de 2, on peut calculer g^n en utilisant $\log_2 n$ opérations au lieu de $n - 1$.

Si $n = \sum_{i=0}^{k} \epsilon_i 2^i$, $\epsilon_i \in \{0, 1\}$, $\epsilon_k = 1$, on a :

$$g^n = \prod_{i=0}^{k} \left(g^{2^i}\right)^{\epsilon_i}$$

$$= \prod_{i=0, \epsilon_i = 1}^{k} g^{2^i}$$

Algorithme 13 Exponentiation binaire

Entrées: g, $n = \sum_{i=0}^{k} \epsilon_i 2^i > 0$, $\epsilon_i \in \{0, 1\}$

Sorties: g^n

1: $\gamma \leftarrow g; \Pi \leftarrow g^0$
2: **Pour** $i = 0$ à k **Faire**
3: **Si** $\epsilon_i = 1$ **Alors**
4: $\Pi \leftarrow \Pi \times \gamma$
5: **Fin si**
6: $\gamma \leftarrow \gamma^2$
7: **Fin pour**
8: **Retourner** Π

Théorème

Cet algorithme calcule g^n en au plus $2(k+1)$ opérations ('$\times$'), soit $O(\log n)$.

Améliorations

$g^{11} = g^8 \times g^2 \times g$

$g^{11} = g \times (g^5)^2 = g \times \left(g \times (g^2)^2\right)^2$

Nouvel algorithme pour calculer g^n

- Si $n = 1$ renvoyer g
- Si n impair renvoyer $g \times \left(g^{\frac{n-1}{2}}\right)^2$
- Sinon renvoyer $\left(g^{\frac{n}{2}}\right)^2$

Prolongements

- Si G est un groupe où l'inverse est "gratuit"
  $n = \sum \epsilon_i 2^i$, $\epsilon_i \in \{0, 1, -1\}$ où le motif $\epsilon_i = \epsilon_{i+1} = 1$ n'apparait pas.
- Fenêtre flexible $g^n = g^\alpha \times \left(g^{\frac{n}{\beta}}\right)^\beta$ où $\alpha = n \mod \beta$ et β est une petite puissance de 2. $\{g^\alpha, 0 \leqslant \alpha < \beta\}$ est précalculé.

2.4 Symbole de Legendre / de Jacobi

2.4.1 p premier impair

Définition

Soit $a \in \mathbb{Z}$.

$$\left(\frac{a}{p}\right) = \begin{cases} 1 & \bar{a} \in \left(\mathbb{F}_p^*\right)^2 \\ 0 & \bar{a} = 0 \\ -1 & \bar{a} \notin \left(\mathbb{F}_p^*\right)^2 \end{cases}$$

Lemme

$\left(\frac{a}{p}\right) = a^{\frac{p-1}{2}} \mod p$ et $\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right) \times \left(\frac{b}{p}\right)$.

Corollaire

$\left(\frac{a}{p}\right)$ se calcule en $O(\log p)$ multiplications dans $\mathbb{F}_p$ et si $0 \leqslant a < p$ en $\tilde{O}(\log_2 p)^2$ opérations élémentaires.

2.4.2 Symbole de Jacobi

$$\left(\frac{a}{2}\right) = \begin{cases} 0 & \text{si a est pair} \\ (-1)^{\frac{a^2-1}{8}} & \text{sinon} = \begin{cases} 1 & \text{si } a = \pm 1 \mod 8 \\ -1 & \text{si } a = \pm 3 \mod 8 \end{cases} \end{cases}$$

Soit $b = \prod_{i=1}^{k} p_i \geqslant 1$, un produit de nombres premiers (pas distincts à priori).
On définit :

$$\left(\frac{a}{b}\right) = \prod_{i=1}^{k} \left(\frac{a}{p_i}\right)$$

Proprits

Si $a, b \in \mathbb{Z}$ sont tels que tous les symboles de Jacobi utilisés sont bien définis, on a :

1. $\left(\frac{ab}{c}\right) = \left(\frac{a}{c}\right)\left(\frac{b}{c}\right)$ et $\left(\frac{a}{bc}\right) = \left(\frac{a}{b}\right)\left(\frac{a}{c}\right)$
2. $a \in ((\mathbb{Z}/b\mathbb{Z})^*)^2 \Rightarrow \left(\frac{a}{b}\right) = 1$

Théorème

Si a, b impairs $\geqslant 1$:

1. $\left(\frac{-1}{b}\right) = (-1)^{\frac{b-1}{2}}$
2. $\left(\frac{2}{b}\right) = \left(\frac{b}{2}\right)$
3. $\left(\frac{a}{b}\right) = \left(\frac{b}{a}\right) \times (-1)^{\frac{a-1}{2}\frac{b-1}{2}} = \begin{cases} -\left(\frac{b}{a}\right) & \text{si } a = b = 3 \mod 4 \\ \left(\frac{b}{a}\right) & \text{sinon} \end{cases}$
4. $\mathbb{Z} \to \{0, 1, -1\}$
   $a \mapsto \left(\frac{a}{b}\right)$ est périodique de période b.

Remarque

Coût de $(-1)^x$: $O(1)$.

Théorème

On suppose que les entiers sont écrits dans une base qui est une puissance de 2. Alors cet algorithme utilise $O(\log(\max(|a|, |b|))^2)$ pour calculer $\left(\frac{a}{b}\right)$.

Algorithme 14 Calcul du symbole de Jacobi

Entrées: $b \geqslant 1$ impair, $a \in \mathbb{Z}$

Sorties: $\left(\frac{a}{b}\right)$ symbole de Jacobi

1: $s \leftarrow 1$
2: **Tant que** $b \neq 0$ **Faire**
3: $\quad s \leftarrow s \times \left(\frac{2}{b}\right)^{v_2(a)}$;
4: $\quad a \leftarrow \frac{a}{2^{v_2(a)}}$;
5: $\quad s \leftarrow (-1)^{\frac{a-1}{2}\frac{b-1}{2}} \times s$;
6: $\quad (a, b) \leftarrow (b \mod a, a)$;
7: **Fin tant que**
8: **Si** $a = 1$ **Alors**
9: $\quad$ **Retourner** s ;
10: **Sinon**
11: $\quad$ **Retourner** 0 ;
12: **Fin si**

Application

Soit $N = pq$, p et q deux premiers distincts.
Je veux prouver que je connais p et q sans divulguer p ou q.

- $\left(\frac{a}{N}\right)$ calculable.
- $\left(\frac{a^2}{N}\right) = 1$ si $PGCD(a, n) = 1$.
- a est un carré dans $(\mathbb{Z}/n\mathbb{Z})^*$ si et seulement si $\left(\frac{a}{p}\right) = \left(\frac{a}{q}\right) = 1$
- On suppose connu $g \in \mathbb{Z}/n\mathbb{Z}$ qui vérifie $\left(\frac{g}{p}\right) = \left(\frac{g}{q}\right) = -1$

Défi : On transmet $h = g^\epsilon u^2$ où $\epsilon \in \{0, 1\}$ est tiré uniformément au hasard, et $u \in \mathbb{Z}/n\mathbb{Z}$. Est-ce un carré ?

Réponse : $\epsilon = 0$ ou $\epsilon = 1$

$\epsilon = 0$ si et seulement si $\left(\frac{h}{p}\right) = 1$

$\epsilon = 1$ si et seulement si $\left(\frac{h}{p}\right) = -1$

2.5 Test de non-primalité sur $\mathbb{Z}$

Théorème

Soit N entier impair. Si N est premier et $0 < a < N$, alors :

1. $a^{N-1} = 1 \mod N$
2. $a^{\frac{N-1}{2}} = \left(\frac{a}{N}\right) \neq 0 \mod N$
3. Soit $N - 1 = 2^e q$, q impair, alors :

$$\begin{cases} a^q = 1 \mod n \\ \exists 0 \leqslant i < e, a^{q^{2^i}} = -1 \end{cases}$$

Algorithme

On tire $a \in]0, N[$ uniformément au hasard. Il y a trois variantes :

(a) Si 1) est faux OU
(b) Si 2) est faux OU
(c) Si (3) est faux

alors succès : N est composé.
Sinon échec. (a) Fermat, (b) Soloway-Strassen, (c) Rabin-Miller

Remarque

Si N est premier : ECHEC inéluctable.

Complexité

La condition 3) se vérifie en calculant $b = a^q \mod n$ puis par au plus e-1 mises au carré.
$O(\log N)$ opérations dans $\mathbb{Z}/n\mathbb{Z}$
$O(\log N)^2$ jacobi pour 2)

Théorème

Si un entier N impair vérifie 3) alors il vérifie 2) et inversement pour $0 < a < N$. On peut construire des paires (N, a) telles que :
1) soit vérifié et pas 2)
1) soit vérifié et pas 3)

Théorème

Soit N impair composé. $\{a \in \mathbb{Z}/n\mathbb{Z}$ tel que $a^{\frac{N-1}{2}} = \left(\frac{a}{N}\right) \neq 0 \mod N\}$ est un sous groupe strict de $(\mathbb{Z}/n\mathbb{Z})^*$.

Théorème de Korselt

Soit N composé.
$\{a \in \mathbb{Z}/n\mathbb{Z}$ tel que $a^{N-1} = 1 \mod N\}$ est un sous-groupe de $(\mathbb{Z}/n\mathbb{Z})^*$.
Il est gal $(\mathbb{Z}/n\mathbb{Z})^*$ si et seulement si :

- N est sans facteur carré
- pour tout premier $p|N$, on a $p-1|N-1$ $(*)$

Définition

Un entier N composé tel que les deux candidats $(*)$ sont vérifiés est dit de Carmichael.

Remarque

Il existe une infinité de tels nombres.

Corollaire

Soit N impair composé.
La probabilité qu'un a tel que $0 < a < N$ vérifie 2) est inférieure ou égale $\frac{1}{2}$.
La probabilité qu'un a tel que $0 < a < N$ vérifie 3) est inférieure ou égale $\frac{1}{2}$.

Remarque

En fait, on démontre que la probabilité qu'un a tel que $0 < a < N$ vérifie 3) est inférieure ou égale $\frac{1}{4}$.

2.6 Test de primalité sur $\mathbb{Z}$

Théorème

N est premier $\Leftrightarrow$ $(\mathbb{Z}/n\mathbb{Z})^*$ a N-1 éléments $\Leftrightarrow$ il existe $g \in (\mathbb{Z}/n\mathbb{Z})^*$ d'ordre N-1.
Soit G un groupe et $\prod p^{e_p} = |G|$ la factorisation de son cardinal alors $g \in G$ est d'ordre n $\Leftrightarrow \begin{cases} g^n = 1 & \text{le neutre} \\ g^{\frac{n}{p}} \neq 1 & \forall p \text{ premier, p divisant n} \end{cases}$.

Corollaire

On peut prouver que $g \in$ est bien d'ordre en utilisant $O(\log n)^2$ multiplications dans le groupe. Calculer $\{g^i, i < n\}$ réclame n multiplications.

Analyse

SI N est premier alors $(\mathbb{Z}/n\mathbb{Z})^*$ est cyclique et a N-1 éléments. Il a donc

Algorithme 15 Preuve de primalité

Entrées: N tel que tous les diviseurs premiers de N-1 soient connus.
Sorties: preuve de primalité de N ou échec.

1: Tirer $1 < a < N$ uniformément au hasard.
2: **Si** a est d'ordre $N-1$ dans $(\mathbb{Z}/n\mathbb{Z})^*$ **Alors**
3: **Retourner** succès
4: **Sinon**
5: **Retourner** échec
6: **Fin si**

$\varphi(N-1)$ générateurs. La probabilité de succès est donc $\geqslant \frac{\varphi(n)}{n}$ où $n = N-1$

$$= \prod_{\substack{p \text{ premier} \\ p|n}} \left(1 - \frac{1}{p}\right) \gg \frac{1}{\log\log n}$$

2.7 Factorisation dans $\mathbb{Z}$

Définitions

Soit R un anneau.

- $R^* = \{\text{units de R}\} = \{x \in R,\ \exists y \in R,\ xy = 1\}$
- $x \in R$ est irréductible si :
  - $x \notin R^*$
  - $x = yz \Rightarrow y \in R^*$ ou $z \in R^*$
- $x \in R$ est un diviseur de zéro si $x \neq 0$ et $\exists y \in R,\ y \neq 0,\ xy = 0$

Définition

R est factoriel si tout élément $x \in R\backslash\{0\}$ s'écrit

$$x = \epsilon \prod_{i=1}^{n} p_i$$

où $\epsilon \in R^*$, p irréductible.

Convention : un quotient vide pour $n = 0$ vaut 1. Et si $x = \epsilon' \prod_{i=1}^{n} p'_i$ est une autre décomposition alors $n = n'$ et il existe $\sigma \in S_n$ tel que p_i et $p'_{\sigma(i)}$ sont associés pour tout i. (A et B sont associés si et seulement si $\exists u \in R^*,\ a = ub$).

Stratégie générale de factorisation si R est euclidien

Soit $x \in R$, si $\bar{a} \in R/(x)$ est un diviseur de zéro alors $PGCD(a, x)$ est un

diviseur de x, ce n'est pas une unité et il n'est pas associé à x. $\Rightarrow x = d\frac{x}{d}$ où $d = PGCD(a, x)$ est une factorisation non triviale.

Soit $N \in \mathbb{Z}$ que l'on veut factoriser. On cherche $\bar{a} \in \mathbb{Z}/n\mathbb{Z}$ un diviseur de zéro.
La variante la plus courante : on cherche $u, v \in \mathbb{Z}$ tels que $u^2 = v^2 \mod N (*) \Rightarrow u = \pm v \mod N \Rightarrow a = u \pm v$ convient.
Supposons $(*)$ avec u, v premiers à $N > 2 \Rightarrow \left(\frac{u}{v}\right)^2 = 1 \mod N$.
L'équation $x^2 = 1$ dans $(\mathbb{Z}/n\mathbb{Z})^*$ a exactement deux solutions
$\Longleftrightarrow (\mathbb{Z}/n\mathbb{Z})^*$ est cyclique.
$\Longleftrightarrow N = 4$ ou $N = p^k$ ou $N = 2p^k$ avec $k \geqslant 1$, p premier impair.

Remarque
Comment tester $N = q^k$? (et calculer q, k ?)

$$k \leqslant \log_q N \leqslant \log_2 N$$

Il suffit de savoir tester si $N = q^k$ pour un k fixé et extraire une racine k-ème.
On a deux méthodes d'analyse numérique pour produire q_n (calculable !) tel que $q_n \longrightarrow N^{\frac{1}{k}}$ dans R.
Dès que $|q_n - N^{\frac{1}{k}}| < \frac{1}{2}$ il n'y a qu'une possibilité pour q :

$$q = \left\lfloor q_n + \frac{1}{k} \right\rfloor = \lfloor q_n \rceil$$

et il suffit de tester si $N = q^k$. Tout ceci se fait en temps $(\log N)^{O(1)}$.

Détails sur l'étape 2)
- Tirer $x \in \mathbb{Z}/n\mathbb{Z}$ uniformément au hasard et calculer $rem(x^2, N)$. S'il se factorise sur $\mathcal{B}$ (division exhaustive) on a une relation sinon on recommence.
- Optimiser B tel que la recherche exhaustive + l'algèbre linéaire ne soient pas trop couteux mais tel qu'on ait beaucoup d'entiers de $[0, N]$ B-friables $\Rightarrow B \approx e^{c(\log N \log\log N)^{\frac{1}{2}}}$
- Choisir J de façon à avoir plus d'inconnues que d'équations pour 3). Il faut juste que $|\mathcal{B}| < J$.

2.8 Factorisation dans $\mathbb{F}_q(\alpha)$

Remarque préliminaire
Soit $T \in k[X]$ avec k un corps, T unitaire, un polynome à factoriser. On

Algorithme 16 Algorithme de Dixon

Entrées: N impair, pas une puissance pure, B une borne de "friabilité"
Sorties: un facteur non trivial de N ou échec.

1: Soit $\mathcal{B} \leftarrow \{p \leqslant B, p \text{ premier}\}$
2: On "produit" des $x_j \in \mathbb{Z}$ avec $j \in J$ tel que $x_j^2 = \prod_{i \in \mathcal{B}} i^{e_{i,j}} \mod N$ avec $e_{i,j} \in \mathbb{Z}$.
3: Soit $(\bar{v_0})$ un élément du noyau de $(\bar{e_{i,j}}) \in M_{|B| \times J}(\mathbb{F}_2)$
$\Leftrightarrow \sum_j e_{i,j} v_j = 0 \mod 2, \forall i.$
4: Alors $\left(\prod_{j \in J} x_j^{v_j} \right)^2 = \sum_{i \in \mathcal{B}} i^{\sum_j e_{i,j} v_j} \mod N$
on pose $\bar{u} = \prod_{j \in J} x_j^{v_j} \mod N$, $\bar{v} = \prod i^{\frac{\sum_j e_{i,j} v_j}{2}} \mod N$ et on a :

$$u^2 = v^2 \mod N$$

5: **Si** $u = \pm v$ **Alors**
6: **Retourner** échec
7: **Sinon**
8: **Retourner** $PGCD(u - v, N)$
9: **Fin si**

peut supposer que T est sans facteur carré (si k est parfait) $\Rightarrow$ car $k = 0$ ou car $k = p$ et $x \to x^p$ surjectif.

Théorème

Soit $\Delta = PGCD(T, T')$

1. $\Delta = 1 \Leftrightarrow T$ est sans facteur carré.
2. $\Delta = T \Leftrightarrow T' = 0 \Leftrightarrow T = t(X^p)$ où $t \in k[X]$ et $p = \text{car } k \Leftrightarrow$(si k est parfait) $T = \tau(X)^p$ où $\tau \in k[X]$.
$$\sum t_i X^{pi} = \left(\sum \tau_i X^i \right)^p \text{ vrai ssi } t_i = \tau_i^p, \ \forall i$$
3. Δ est un facteur strict de T. On peut étudier séparément Δ et $\frac{T}{\Delta}$ et $\frac{T}{\Delta}$ est sans facteur carré.

Algorithme de Berlekamp

Soit $T \in \mathbb{F}_q[X]$, unitaire sans facteur carré. On a $T = \prod_{i=1}^{s} T_i$ avec T_i

irréductibles unitaires.

$$A = \mathbb{F}_q[X]/(T) \simeq \prod_{i=1}^{s} \mathbb{F}_q[X]/(T_i) \simeq \prod_{i=1}^{s} \mathbb{F}_q \times \deg T_i$$

$$B = \text{Ker} \begin{pmatrix} A & \longrightarrow & A \\ x & \mapsto & x^q - x \end{pmatrix}$$

Corollaire

T irréductible $\Leftrightarrow \dim \mathbb{F}_q \times B = 1$

$\dim B = \#$ facteurs carrés irréductibles distincts de T.

Soit $\phi : \begin{matrix} A & \longrightarrow & A \\ x & \mapsto & x^q \end{matrix}$. On veut calculer ker $(\phi - Id)$. On écrit la matrice de ϕ dans la base $(\bar{1}, \bar{X}, \bar{X^2}, \ldots, X^{(\bar{deg}T)-1})$.

$\phi(\bar{X_i}) = \bar{X^{qi}} = (\bar{X^i})^q$ par exponentiation binaire.

Mieux : $\phi(\bar{1}) = 1$, $\phi(\bar{X}) = \bar{X^q}$ par exponentiation binaire, $\phi(X^{i+1}) = \phi(X^i)\phi(X)$: par tout a on peut écrire la matrice de $\phi - Id$.

Corollaire

On a un algorithme déterministe polynomiale en $\log q$ et T.

On cherche un diviseur de zéro dans B donc dans A. Soit $\bar{x} \in B$, $x \notin \mathbb{F}_q$. Il existe $\alpha \in \mathbb{F}_q$ tel que $x - \alpha$ soit diviseur de zéro.

Algorithme : Pour tout $\alpha \in \mathbb{F}_q$, tester si $PGCD(x - \alpha, T)$ est non-trivial.

Maintenant q est impair.

Idée : $\bar{x} \in B$, $\bar{y} = \bar{x}^{\frac{q-1}{2}}$. Les composantes de y dans $\prod_{i=1}^{s} \mathbb{F}_q$ sont dans $\{-1, 0, 1\}$.

$PGCD(y - 1, T)$?

Mauvais cas :

- $-\left(\frac{q-1}{2}\right)^s$ [tous les $y_i = 1$]
- $-\left(\frac{q+1}{2}\right)^s$ [tous les $y_i \neq 1$]

Chapitre 3

Systèmes polynomiaux

Motivation

Soit k un corps et $f_i \in k[X_1, \ldots, X_n]$

$$(S) \begin{cases} f_1(x_1, \ldots, x_n) = 0 \\ \vdots \\ f_m(x_1, \ldots, x_n) = 0 \end{cases}$$

$x_i \in k$?

$$< f_1, \ldots, f_m > \subseteq k[X_1, \ldots, X_n]$$
$$I \subseteq k[X_1, \ldots, X_n]$$
$$f \in I \,?$$

3.1 Ordres monomiaux sur $k[x_1, \ldots, x_n]$, k corps

Notation : $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n$
$X^\alpha = X_1^{\alpha_1} \ldots X_n^{\alpha_n}$ est un monome.

Remarques

1. Les monomes sont en bijection avec $\mathbb{N}^n$
2. Tout ordre sur $\mathbb{N}^n$, $\alpha < \beta$ définit un ordre sur les monomes $X^\alpha < X^\beta$

Exemple : ordre lexicographique (strict) noté lex

$(\alpha_i) < (\beta_i) \Longleftrightarrow (\alpha_j < \beta_j$ si j est la première coordonnée où (α_i) et (β_i) diffèrent.

Définition
Un ordre est monomial si :
- ordre total ($\alpha < \beta$ ou $\alpha > \beta$ ou $\alpha = \beta$)
- $\alpha > \beta$ et $\gamma \in \mathbb{N}^n \Rightarrow \alpha + \gamma > \beta + \gamma$
- $\alpha \in \mathbb{N}^n \Rightarrow \alpha \geqslant 0 = (0, \ldots, 0)$

Théorème
lex est monomial.

On fixe un ordre pour la suite.

Définitions
Soit $f = \sum_{\alpha \in \mathbb{N}^n} a_\alpha X^\alpha \neq 0$ dans $k[x_1, \ldots, x_n]$ avec $a_\alpha \in k$ et $a_\alpha = 0$ pour presque tout α.

1. $\deg f = \max\{\alpha \in \mathbb{N}^n,\ a_\alpha \neq 0\}$ [multidegré de $\mathbb{N}^n$]
2. $cd(f) = a_{\deg f} \in k$
3. monome dominant : $X^{\deg f}$ noté $md(f)$
4. terme dominant : $a_{\deg f} X^{\deg f}$ noté $td(f)$

Lemme
Soient $f, g \neq 0$ (dans $k[x_1, \ldots, x_n]$). On a :
$\deg(fg) = \deg f + \deg g$ (somme composante par composante)
$\deg(f + g) \leqslant \max(\deg f, \deg g)$ si $f + g \neq 0$

3.2 Pseudo division euclidienne

Théorème
Soit $(f_1, \ldots, f_s) \in k[x_1, \ldots, x_n]$
Tout $f \in k[x_1, \ldots, x_n]$ s'écrit :

$$f = \sum_{i=1}^{s} q_i f_i + r$$

où $r, q_i \in k[x_1, \ldots, x_n]$, avec $r = 0$ ou $r = \sum r_\alpha X^\alpha$ tel qu'aucun des monomes X^α ($r_\alpha \neq 0$) ne soit divisible par l'un des $md(f_i)$.

Définition
r est un reste de la division de f par $(f_1, \ldots, f_s)$

Algorithme 17 Algorithme division euclidienne

Entrées: f, f_i
Sorties: q_i, r

$q_1 \leftarrow 0$;
$\vdots$
$q_s \leftarrow 0$;
$r \leftarrow 0$;
$p \leftarrow f$;
Tant que $p \neq 0$ **Faire**
  $i \leftarrow 1$; $div \leftarrow faux$;
  Tant que $i \leqslant s$ et $div = faux$ **Faire**
    Si $md(f_i)|md(p)$ **Alors**
      $q_i \leftarrow q_i + \frac{td(p)}{td(f_i)}$;
      $p \leftarrow p - \frac{td(p)}{td(f_i)} f_i$;
      $div \leftarrow vrai$;
    Sinon
      $i \leftarrow i + 1$;
    Fin si
  Fin tant que
  Si $div = faux$ **Alors**
    $r \leftarrow r + td(p)$;
    $p \leftarrow p - td(p)$;
  Fin si
Fin tant que
Retourner q_i, r

3.3 Résultats, Applications

Notation

Soit $(g_\lambda)_{\lambda \in E}$ est un ensemble de polynomes de $k[x_1, \ldots, x_n]$. On note $< g_\lambda >$ le plus petit idéal de $k[x_1, \ldots, x_n]$ contenant tous les g_λ.

Définition

Soit I un idéal de $k[x_1, \ldots, x_n]$. Un système $(g_1, \ldots, g_s)$ de générateurs de I est une base de Grobner si et seulement si pour tout $f \in k[x_1, \ldots, x_n]$ le reste de la division euclidienne de f par $(g_1, \ldots, g_s)$ est bien défini. Il existe un unique r tel que $f = \sum_{i=1}^{s} q_i g_i + r$.

Théorème
A partir d'un système fini de générateurs de I, il existe un algorithme qui calcule une base de Grobner de I.

Il y a une notion de base canonique, tout idéal I admet une unique base de Grobner "réduite".

Corollaire
Si $f \in k[x_1, \ldots, x_n]$, on a $f \in I \Leftrightarrow r = 0$ (reste de la division par base de Grobner)

Corollaire
On peut tester si $I = J$ ($\Leftrightarrow$ les bases de Grobner réduites sont égales).

Définition équivalente
$g_1, \ldots, g_s$ est une base de Grobner de I $\Leftrightarrow$

$$< md(g_i) >_{i=1,\ldots,s} =< md(g), g \in I >$$

3.4 Résoudre les systèmes d'équations polynomiales

Définition
Soit $I =< f_1, \ldots, f_s > \in k[x_1, \ldots, x_n]$
Le l-ème idéal d'élimination I_l est l'idéal de $k[x_{l+1}, \ldots, x_n]$:

$$I_l = I \cap k[X_{l+1}, \ldots, X_n]$$

Théorème
Soit G une base de Grobner de I pour lex. Alors $G_l = G \cap k[X_{l+1}, \ldots, X_n]$ est une base de Grobner de I_l.

Corollaire
Si $f_1(x_1, \ldots, x_n) = \ldots = f_s(x_1, \ldots, x_n) = 0$ alors on a :

$$g_n(x_n) = 0$$

$$g_{n-1}(x_n, x_{n-1}) = 0$$

$$g_{n-2}(x_n, x_{n-1}, x_{n-2}) = 0$$

$$\vdots$$

"forme triangulaire"
pour des g_i calculables.

Corollaire

Si :

$$(E)\begin{cases} x_1 = f_1(t_1, \ldots, t_m) \\ \vdots \\ x_n = f_n(t_1, \ldots, t_m) \end{cases}$$

est un système d'équations paramétriques, on fixe $t_1 > t_2 > \ldots > t_m > x_1 > \ldots > x_n$. Alors I_m définit le plus petit ensemble (défini par des équations polynomiales) contenant (E).

Théorème

$I =< f_1, \ldots, f_s > \subset \mathbb{C}[X_1, \ldots, X_n]$

1. Le système $(S)\begin{cases} f_1(x_1, \ldots, x_n) = 0 \\ \vdots \\ f_s(x_1, \ldots, x_n) = 0 \end{cases}$ a une solution $(x_1, \ldots, x_n) \in \mathbb{C}^n$ si et seulement si la base de Grobner réduite de I est $\neq \{1\}$.
2. (S) a un nombre fini de solutions si et seulement si $\forall i \leqslant n$ une puissance de x_i est dans $< md(f), f \in I >$.