Indice

Capitolo 1

Introduzione all'Algebra Computazionale

Gli algoritmi che verranno trattati si divideranno in due classi. Vedremo infatti gli algoritmi classici, come per esempio l'algoritmo di divisione su un anello euclideo. Tali algoritmi sono caratterizzati dalla proprietà di risolvere il problema in maniera diretta, lavorando sugli oggetti forniti in input. Gli algoritmi moderni risolvono invece il problema in ambiti diversi da quelli naturali e ritrasportano indietro la soluzione. Un esempio è fornito dal caso di $\mathbb{Z}$ e dei quozienti $\mathbb{F}_p$; spesso è più agevole lavorare su questi quozienti e ricostruire la soluzione in $\mathbb{Z}$.

1.1 Operazioni elementari e rappresentazioni

Lavoreremo principalmente con polinomi e con i cosiddetti "oggetti effettivi":

Definizione 1.1. Un *anello effettivo* è un insieme in cui ogni elemento può essere descritto da una sequenza finita di simboli appartenenti ad un alfabeto finito, e in cui sia possibile definire la somma e il prodotto, e decidere se due elementi sono uguali

In base a questa definizione, $\mathbb{R}$ non è per esempio effettivo, poiché il numero di sequenze finite in un alfabeto finito sono numerabili, mentre i numeri reali sono più che numerabili. Gli oggetti che tratteremo saranno per lo più anelli di polinomi su un anello effettivo e l'anello $\mathbb{Z}$. Vediamo quindi come rappresentare un polinomio o un intero. Un polinomio

$$p(x) = \sum_{i=1}^{n} a_i x^i$$

lo rappresenteremo solitamente come un array $(a_0, \ldots, a_n)$. Per esempio, il polinomio $x^2 + x + 2$ è rappresentato dalla tripla $(2, 1, 1)$. Tale rappresentazione non è sempre ottimale; se infatti il polinomio fosse di grado alto con molti coefficienti nulli ("sparso"), si occuperebbe molta memoria per salvare in realtà poche informazioni. Si può allora adottare una rappresentazione sparsa:

$$((a_i, d_i))$$

dove a_i è il coefficiente dell'i-esimo termine non nullo e d_i è il grado. Per esempio, con questa rappresentazione $x^{1000} - x + 1$ sarebbe $((1, 1000), (-1, 1), (1, 0))$, più maneggevole di un array con 1001 elementi.
La rappresentazione degli interi è in pratica equivalente a quella dei polinomi. Poiché infatti l'implementazione data nei principali linguaggi non permette di lavorare con numeri troppo grandi, si preferisce scegliere B una base adeguata e rappresentare un intero come un array

$$x = \sum_{i=1}^{n} b_i B^i \qquad\qquad x = (b_1, \ldots, b_n)$$

Vediamo ora come effettuare le operazioni più semplici su questi anelli: consideriamo dapprima il caso dei polinomi. Consieriamo

$$a(x) = \sum_{i=0}^{n} a_i x^i \qquad\qquad b(x) = \sum_{i=0}^{m} b_i x^i$$

Supponiamo per semplicità $n \geq m$ Sappiamo che la somma è data da

$$c(x) = a(x) + b(x) = \sum_{i=0}^{n} (a_i + b_i) x^i$$

e dunque per calcolare la somma è sufficiente calcolare la somma di n elementi. Dunque la complessità dell'addizione è

$$A(m, n) = O(n)$$

Per quanto riguarda la moltiplicazione, sappiamo che i coefficienti del prodotto $c(x) = a(x) \cdot b(x)$ sono dati da

$$c_k = \sum_{i=0}^{k} a_i b_{k-i}$$

e dunque per il k-esimo termine dobbiamo calcolare k prodotti e k somme. Sia ha allora che la complessità della moltiplicazione è

$$M(m, n) = O(mn)$$

Vediamo ora la divisione. Vogliamo cioè trovare $q(x), r(x)$ tali che $a(x) = q(x)b(x) + r(x)$ in modo tale che $r = 0$ o $deg(r) < deg(b)$. Ad ogni passo, dobbiamo moltiplicare $b(x)$ per a_i/b_m e per un'opportuna potenza di x e poi sottrarre il risultato ad $a_i(x)$, dove a_i è il polinomio ottenuto in output dall'$i-1$-esimo passo. Dunque, visto che il numero di iterazioni è al più $m - n + 1$, si ottiene

$$D(n, m) = O(n(n - m))$$

Su $\mathbb{Z}$, quanto detto fino ad ora è facilmente generalizzabile. L'unico problema è la gestione del riporto: bisogna cioè mantenere la rappresentazione in base B. Questo può essere fatto facilmente durante l'esecuzione dell'algoritmo, senza rendere più onerosi i procedimenti descritti.

Vediamo ora il calcolo del massimo comune divisore (gcd) su $\mathbb{Z}$. Sappiamo che il calcolo di $gcd(a, b)$ può essere svolto notando che, da una relazione $a =$

$bq + r$ si ha $gcd(a, b) = gcd(b, r)$. Il gcd si calcola quindi dall'algoritmo euclideo. Indicizzando $s_0 = a$, $s_1 = b$, si ha la relazione $s_{i-1} = q_i s_i + s_{i+1}$. L'algoritmo termina quando il resto è 0, cioè $s_t = 0$. La terminazione viene garantita dalla funzione grado che decresce ad ogni iterazione: a priori sono allora necessari $a+b$ passi per la terminazione. Un'analisi più attenta fornisce il seguente risultato, che può essere generalizzato anche al caso di polinomi:

Proposizione 1.2. Sia $N \geq a > b > 0$. Allora l'algoritmo euclideo per il calcolo di $gcd(a, b)$ termina in $O(\log N)$ iterazioni.

Dimostrazione. Dall'iterazione generica, otteniamo

$$s_{i-1} = s_i q_i + s_{i+1} > s_{i+1}(1 + q_i)$$

Chiamando d il numero di iterazioni (ossia $s_{d+1} = 0$ e $s_d > 0$), poniamo che $d > 1$ e che $d - 1 \leq 2k \leq d$. Avremo

$$a = s_0 > s_2(1 + q_1) > s_4(1 + q_3)(1 + q_1) > \cdots > s_{2k} \prod_{i=1}^{k} (1 + q_{2i-1})$$

Dato che $q_i, s_i \geq 1$ per ogni $i \leq d$, otteniamo

$$N \geq a > s_{2k} \prod_{i=1}^{k} (1 + q_{2i-1}) \geq 2^k \geq 2^{\frac{d-1}{2}} \implies d \leq 2 \log_2 N + 1 = O(\log N)$$

□

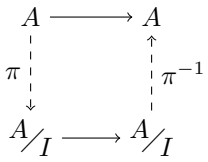
Considerando a, b vicini a N, e ricordandoci che la complessità di una divisione è $O(\log(a)(\log(a) - \log(b)))$, otteniamo che la complessità totale di un gcd tra interi è $O((\log N)^3)$.

Esercizio. Per i polinomi a coefficienti razionali, il gcd fatto con il metodo di Euclide ha una complessità di $O(N^2)$, dove N è il massimo tra i gradi dei due polinomi

L'algoritmo euclideo non solo determina il gcd ma può anche fornire i coefficienti dell'identità di Bezout. Dati $a, b \in \mathbb{Z}$, detto $d = gcd(a, b)$, esistono $s, t \in \mathbb{Z}$ tali che $as + bt = d$. Per trovarli, è sufficiente memorizzare le operazioni compiute nell'algoritmo. Per esempio:

a	b	d	
1	0	a	
0	1	b	
1	$-q_1$	r_1	$a = bq_1 + r_1$
$-q_2$	$1 + q_1 q_2$	r_2	$b = q_2 r_1 + r_2$

e così via. Il problema di questo algoritmo è che polinomi con coefficienti bassi possono generare durante l'algoritmo una crescita notevole dei coefficienti e dunque rallentare il calcolo. Si preferisce allora a questo algoritmo classico un algoritmo moderno, chiamato **Schema per Immagine Omomorfa**: si lavora quindi su anelli quoziente e si riporta poi la soluzione sull'anello di partenza.



Il primo problema che si pone in questo tipo di approccio è la commutatività del diagramma:

Definizione 1.3. Gli ideali $I \subseteq A$ per i quali il diagramma commuta si dicono *lucky* o *fortunati*.

Un altro problema è se il quoziente è di un anello effettivo sia ancora effettivo. Questo accade se disponiamo di un insieme di rappresentanti. Nel caso di $A = \mathbb{Z}$ e un ideale $I = (m)$, possiamo scegliere due rappresentazioni:

- La rappresentazione *positiva*: $[0], [1], \ldots, [m-1]$
- La rappresentazione *bilanciata*: $[-\frac{m}{2} + 1], \ldots, [\frac{m}{2}]$

Per $m = 5$, per esempio, abbiamo le due rappresentazioni

$$[0], [1], [2], [3], [4] \qquad\qquad [-2], [-1], [0], [1], [2]$$

L'idea è quella di lavorare con primi di $\mathbb{Z}$ molto grandi in modo che il rappresentante ottenuto come soluzione nel quoziente coincida con la soluzione in A. Un ideale I che ha questa proprietà viene detto *ampio*. Si utilizzano solitamente due tecniche:

- Teorema cinese del resto: si trovani ideali $I_1, \ldots, I_n$ a due a due comassimali tali che $I_1 \cap \ldots, \cap I_n$ sia ampio
- Dato un ideale I, si ricostruisce la soluzione modulo I^n in modo tale che I^n sia ampio

Analizziamo la prima soluzione:

Teorema 1.4 (Cinese del Resto)**.** Siano $I_1, \ldots, I_n \subseteq A$ ideali a due a due comassimali e siano $a_1, \ldots, a_n \in A$. Allora esiste $a \in A$ tale che $a \equiv a_i \ (I_i)$ per ogni I_i. Inoltre, a è unico modulo $I_1 \cap \cdots \cap I_n = \prod_{i=1}^{n} I_i$.

Dimostrazione. Per comassimalità, presi comunque $i \neq j$, esistono elementi tali che

$$\underbrace{\eta_i^{(j)}}_{\in I_i} + \underbrace{\eta_j^{(i)}}_{\in I_j} = 1$$

Definiamo allora

$$L_i \coloneqq \prod_{j \neq i} \eta_j^{(i)}$$

Notiamo che $L_i \equiv 0 \ (I_j)$ e $L_i \equiv 1 \ (I_i)$, poiché $\eta_j^{(i)} \in I_j$, e

$$\eta_i^{(j)} \in I_i \implies L_i = \prod_{j \neq i} \eta_j^{(i)} \equiv \prod_{j \neq i} \left(\eta_j^{(i)} + \eta_i^{(j)} \right) = 1 \pmod{I_i}$$

Allora $a = \sum_{i=1}^{n} a_i L_i$ è l'elemento cercato. Se esistesse un'altra soluzione b, allora la differenza $a - b$ starebbe in tutti gli I_i, dunque a è l'unica soluzione modulo $I_1 \cap \cdots \cap I_n$. □

La soluzione fornita nella dimostrazione è di tipo Lagrangiano: richiede cioè prima il calcolo di ogni L_i per poter trovare a. Esiste però anche un approccio Newtoniano, ricorsivo, conveniente per quanto riguarda la complessità. Dati $a_1, \dots, a_n$, poniamo $a^{(1)} = a_1$. Supponiamo $a^{(i-1)} \equiv a_j(I_j)$ per ogni $j < i$ e poniamo l'errore i-esimo $e_i = a_i - a^{(i-1)}(I_i)$. Definiamo allora

$$a^{(i)} = a^{(i-1)} + e_i \underbrace{\prod_{j=1}^{i-1} \eta_j^{(i)}}_{q_i} \quad \forall\, i > 1$$

Avremo che

$$a^{(i)} \equiv a^{(i-1)} \equiv a_j \pmod{I_j} \quad \forall\, j < i \qquad a^{(i)} \equiv a^{(i-1)} + e_i = a_i \pmod{I_i}$$

Ponendo $e_1 = a_1$ e $q_1 = 1$, otteniamo dopo n iterazioni che

$$a = a^{(n)} = a^{(n-1)} + e_n q_n = a^{(n-2)} + e_{n-1} q_{n-1} + e_n q_n = \cdots = \sum_{i=1}^{n} e_i q_i$$

1.2 Interpolazione

Ci poniamo ora il problema, dati $\alpha_1, \dots, \alpha_n$ e $\beta_1, \dots, \beta_n$ di trovare un polinomio f tale che $f(\alpha_i) = \beta_i$, cioè

$$\begin{cases} f(\alpha_1) = \beta_1 \\ f(\alpha_2) = \beta_2 \\ \vdots \\ f(\alpha_n) = \beta_n \end{cases}$$

Possiamo vedere questo problema anche sui quozienti:

$$\begin{cases} f(x) = \beta_1 \pmod{(x - \alpha_1)} \\ f(x) = \beta_2 \pmod{(x - \alpha_2)} \\ \vdots \\ f(x) = \beta_n \pmod{(x - \alpha_n)} \end{cases}$$

In questo modo, il problema diventa facilmente risolubile in base al Teorema Cinese del Resto. Notiamo che, dato che gli ideali in considerazione sono principali, risulta anche molto facile trovare i coefficienti $\eta_i^{(j)}$ della dimostrazione. Infatti, se

$$k_i^{(j)}(x) * (x - \alpha_i) + k_j^{(i)}(x) * (x - \alpha_j) = 1$$

valutando in α_j, otteniamo

$$k_i^{(j)}(\alpha_j) * (\alpha_j - \alpha_i) = 1 \implies k_i^{(j)} = (\alpha_j - \alpha_i)^{-1} \qquad k_j^{(i)} = -(\alpha_j - \alpha_i)^{-1}$$

$$\eta_i^{(j)} = \frac{x - \alpha_i}{\alpha_j - \alpha_i} \qquad \eta_j^{(i)} = \frac{x - \alpha_j}{\alpha_i - \alpha_j}$$

e dunque possiamo facilmente trovare il polinomio interpolante. Diamo ora lo pseudocodice dell'algoritmo Newtoniano con complessità $O(N^2)$. Supporremo

quindi di lavorare su un anello euclideo $F[x]$; cerchiamo $f \in F[x]$ tale che $f(\alpha_i) = \beta_i$ per ogni $i = 0, \ldots, N$.

$f(x) = \beta_0$
$M(x) = 1$
for $k = 1, \ldots, N$ **do**
    $M(x) = M(x) \cdot (x - \alpha_{k-1})$
    $c = M(\alpha_k)^{-1}$
    $e = (\beta_k - f(\alpha_k)) \cdot c$
    $f(x) = f(x) + e \cdot M(x)$
end for
return $f(x)$

Per questo algoritmo abbiamo però necessità di saper valutare un polinomio. Questo può essere fatto efficientemente con il metodo di Horner. Supponiamo di voler valutare il polinomio $f = \sum a_i x^i$ nel punto α. Otteniamo allora:

$v = a_n$
for $i = 1, \ldots, n$ **do**
    $v = v \cdot \alpha$
    $v = v + a_{n-i}$
end for
return v

La valutazione richiede quindi $O(n)$ addizioni e $O(n)$ moltiplicazioni. Valutare un polinomio in N punti costa quindi $N^2 + O(n)$. In realtà, in caso di particolari simmetrie dei punti in cui si vuole valutare il polinomio si può far meglio:

Definizione 1.5. Un insieme E_N di cardinalità N si dice simmetrico se

$$\alpha \in E_N \Rightarrow -\alpha \in E_N$$

Proposizione 1.6. Se E_N è simmetrico e $N = 2n$, per valutare un polinomio $f(x)$ di grado $2n$ servono $\frac{N^2}{2} + O(n)$ operazioni.

Dimostrazione. È sufficiente notare che possiamo spezzare il polinomio

$$f(x) = \sum_{i=1}^{2n} a_i x^i = \underbrace{\sum_{i=0}^{n} a_{2i} x^{2i}}_{b(x^2)} + x \underbrace{\sum_{i=0}^{n-1} a_{2i+1} x^{2i}}_{c(x^2)}$$

Per valutare f è allora sufficiente calcolare i polinomi di grado rispettivamente n e $n-1$ in α^2; il numero di operazioni è così dell'ordine $2 \cdot \left(\frac{N^2}{4} + O(n)\right)$. □

Ma si può essere ancora più furbi. Supponiamo che nel campo vi siano delle radici dell'unità. Data ω una radice primitiva N-esima dell'unità, consideriamo

$[\omega] = \{1, \omega, \ldots, \omega^{N-1}\}$, con N pari. Sappiamo che valgono le seguenti proprietà:

$$\omega^2 \text{ è una radice } \frac{N}{2} = n\text{-esima primitiva dell'unità} \qquad \omega^{k+n} = -\omega^k$$

Possiamo allora valutare ricorsivamente un polinomio di grado $N-1 = 2^m - 1$ nelle radici N-esime dell'unità con la cosiddetta FFT. L'algoritmo prende in input il numero dei coefficienti N del polinomio (grado più uno), il vettore dei coefficienti del polinomio $a = \sum a_i x^i$ e una radice N-esima dell'unità e restituisce un vettore $R = (r_0, \ldots, r_{N-1})$ tale che $r_i = a(\omega^i)$.

FFT$(N, (a_0, \ldots, a_{N-1}), \omega)$
if $N = 1$ **then**
  $r_0 = a_0$
else
  $n = \frac{N}{2}$
  $b = (a_0, a_2, \ldots, a_{2n-2})$
  $c = (a_1, a_3, \ldots, a_{2n-1})$
  $B =$ **FFT**(n, b, ω^2)
  $C =$ **FFT**(n, c, ω^2)
  for $k = 0, \ldots, n-1$ **do**
    $r_k = B_k + \omega^k C_k$
    $r_{k+n} = B_k - \omega^k C_k$
  end for
end if
return $(r_0, \ldots, r_{N-1})$

L'algoritmo termina sicuramente, perchè il grado dei polinomi nella chiamata ricorsiva diminuisce ad ogni passo. Dimostriamo la correttezza per induzione sul grado del polinomio. Se $N = 1$, $\deg(a) = 0$ e dunque è il caso base della ricorsione, cioè $r = (a_0)$. Mostriamo ora il passo induttivo. Chiaramente

$$a(x) = b(x^2) + xc(x^2)$$

Poiché $B_k = b(\omega^{2k})$ e $C_k = c(\omega^{2k})$ si ha, se $k < n$, $r_k = B_k + \omega^k C_k = b(\omega^{2k}) + \omega^k c(\omega^{2k}) = a(\omega^k)$. La verifica per il caso $k \geq n$ è analoga, da cui la correttezza.

Soffermiamoci ora ad esaminare il costo computazionale dell'algoritmo. Ad ogni passaggio le uniche moltiplicazioni che operiamo sono $\omega^k * C_k$, poiché supponiamo di avere i valori delle radici dell'unità a disposizione. Dunque operiamo $\frac{N}{2}$ moltiplicazioni, portandoci alla relazione

$$\begin{aligned} M(N) =& 2M\left(\frac{N}{2}\right) + \frac{N}{2} \\ =& 2\left(M\left(\frac{N}{4}\right) + \frac{N}{4}\right) + \frac{N}{2} \\ =& 2^m M(1) + m2^{m-1} \\ \sim& \frac{N}{2}\log N \end{aligned}$$

Definizione 1.7. Siano $\alpha_0, \ldots, \alpha_{N-1} \in F$, definiamo la matrice di Vandermonde degli α_i come

$$V(\alpha_0, \ldots, \alpha_{N-1}) = \begin{pmatrix} 1 & \alpha_0 & \ldots & \alpha_0^{N-1} \\ 1 & \alpha_1 & \ldots & \alpha_1^{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha_{N-1} & \ldots & \alpha_{N-1}^{N-1} \end{pmatrix}$$

Proposizione 1.8. Sia $a(x) = \sum_{i=0}^{N-1} a_i x^i \in F[x]$ e consideriamo il vettore dei coefficienti di a $\underline{a} = (a_0, \ldots, a_{N-1})$. Siano $\alpha_0, \ldots, \alpha_{N-1} \in F$ e sia $V = V(\alpha_0, \ldots, \alpha_{N-1})$ la rispettiva matrice di Vandermonde. Allora

$$V\underline{a} = \underline{b} \iff b_i = a(\alpha_i)$$

Dimostrazione. Basta svolgere il prodotto. □

Tale proposizione lega il problema dell'interpolazione a quello della valutazione. In realtà, permette anche di trovare l'inversa di una matrice di Vandermonde in $O(N^2)$[1], risultato notevole dato che una generica matrice richiede $O(N^3)$ operazioni. Se inoltre la matrice di Vandermonde è costruita sulle radici primitive dell'unità, sappiamo a priori la forma della sua inversa:

Proposizione 1.9. Se V è la matrice di Vandermonde definita da

$$V = V(1, \omega, \omega^2, \ldots, \omega^{n-1})$$

con ω radice n-esima primitiva dell'unità, in un campo in cui $n \neq 0$, allora

$$V^{-1} = \frac{1}{n} V\left(1, \omega^{-1}, \omega^{-2} \ldots, \omega^{-(n-1)}\right)$$

[1] perché? Per trovare la controimmagine di un elemento della base canonica richiede l'interpolazione su N punti, che va in N^2, dunque in tutto N^3..

Capitolo 2

GCD e Fattorizzazione

2.1 Calcolo del GCD di due polinomi: un algoritmo moderno

Siano $f, g \in \mathbb{Q}[x]$. Vogliamo trovare un algoritmo che permetta il calcolo del gcd diverso dall'algoritmo euclideo, per evitare la crescita dei coefficienti che intervengono nell'esecuzione. Lavorare su $\mathbb{Q}$ risulta particolarmente oneroso: i razionali sono infatti implementati come una coppia di interi e sono quindi poco maneggevoli. D'altronde, per il lemma di Gauss possiamo ridurci a lavorare su $\mathbb{Z}$ a meno di moltiplicare per i denominatori. Chiaramente, il primo passo è a questo punto rendere il polinomio primitivo: vogliamo cioè ridurci a due polinomi $f, g \in \mathbb{Z}[x]$ primitivi. Ancora una volta, l'idea è quella di lavorare su quozienti per primi di $\mathbb{Z}$

$$\pi_p \colon \mathbb{Z}[x] \longrightarrow {}^{\mathbb{Z}}\!/_{p\mathbb{Z}}$$

Chiamiamo $d = (f, g)$, $d_p = \pi_p(d)$, $f_p = \pi(f)$, $g_p = \pi(g)$ e sia $D = (f_p, g_p)$. Bisogna trovare un insieme ampio di primi in modo tale da ottenere la soluzione nei quozienti e sollevarla a $\mathbb{Z}[x]$. Per questo, il primo problema è evitare di scegliere dei primi che dividono il coefficiente direttore di d.

Proposizione 2.1. Sia $p \in \mathbb{Z}$ un primo e supponiamo $p \nmid lc(d)$. Allora $\deg(d_p) = \deg d \leq \deg(D)$.

Dimostrazione. Per definizione di gcd, abbiamo

$$f = d\tilde{f} \qquad g = d\tilde{g} \qquad (\tilde{f}, \tilde{g}) = 1$$

Di conseguenza, nel quoziente, $f_p = d_p\tilde{f}_p$ e $g_p = d_p\tilde{g}_p$ e dunque $d_p \mid (f_p, g_p) = D \implies deg(d_p) = deg(d) \leq deg(D)$ □

Notiamo che in generale non vale l'uguaglianza di grado: per esempio, sia $f = (x+1)(x+4)$ e sia $g = (x+1)^2$. Su $\mathbb{F}_3$, si ha allora $(f, g) = (x+1)^2$ mentre su $\mathbb{Z}$ si ha $(f, g) = (x+1)$.
Sappiamo di certo però che il grado del gcd nel quoziente non può diminuire scegliendo $p \nmid lc(d)$. Ricordiamo dunque alcune proprietà del risultante:

$$f, g \in \mathbb{Z} \implies \mathrm{Ris}(f, g) \in \mathbb{Z} \qquad (f, g) \neq 1 \implies \mathrm{Ris}(f, g) = 0$$

Queste portano al seguente risultato.

Proposizione 2.2. Sia $p \in \mathbb{Z}$ un primo tale che $p \nmid u = gcd(lc(f), lc(g))$, $p \nmid \mathrm{Ris}(\tilde{f}, \tilde{g})$, $p \nmid lc(\tilde{f})$ e $p \nmid lc(\tilde{g})$. Allora d_p è associato a D.

Dimostrazione. Sicuramente, $lc(d)|u$, dunque $p \nmid lc(d)$, e per quanto visto sopra, $d_p|D = (f_p, g_p)$, quindi basta mostrare che hanno lo stesso grado. Se per assurdo $(\tilde{f}_p, \tilde{g}_p) \neq 1$, allora avremo che

$$0 = \mathrm{Ris}(\tilde{f}_p, \tilde{g}_p) = \mathrm{Ris}(\tilde{f}, \tilde{g})_p$$

ma questo è assurdo in quanto $p \nmid \mathrm{Ris}(\tilde{f}, \tilde{g})$. Allora, $(\tilde{f}_p, \tilde{g}_p) = 1$ da cui la tesi. □

Questo ci dice che, a meno di finiti primi, possiamo operare l'algoritmo di gcd su $\mathbb{F}_p$, e ottenere un associato di d con il suo stesso grado. Inoltre, se $d = (f, g)$, allora $lc(d)|u$, e chiamato d_k il coefficiente direttore di d, avremo $cd_k = u$. Ponendo di aver preso un primo abbastanza grande, e chiamando $d_p = (f_p, g_p)$ il gcd monico, avremo che, ponendo il tutto in notazione bilanciata,

$$ud_p = cd_k d_p = cd$$

Dato che prendiamo f e g primitivi, anche d lo sarà, dunque prendendo la parte principale di ud_p otteniamo proprio d.
Diamo ora lo pseudocodice dell'algoritmo. Supponiamo di avere in input $f, g \in \mathbb{Z}[x]$ primitivi; restituiremo in output gcd $(f, g) \in \mathbb{Z}[x]$. Supponiamo inoltre di sapere u e un bound B sui coefficienti del gcd.

```
Scegliere p_1 ∈ ℤ tale che p_1 ∤ u
d_1 = (f_{p_1}, g_{p_1}) ∈ 𝔽_{p_1}[x] monico
if deg d_1 = 0 then
    (f, g) = 1
else
    d_1 = u d_1 ∈ 𝔽_{p_1}[x] in notazione bilanciata
    Scegliere p_2 ∈ ℤ primo tale che p_2 ∤ u
    d_2 = (f_{p_2}, g_{p_2}) ∈ 𝔽_{p_2} monico
    if deg d_2 > deg d_1 then
        Scegliere un altro primo p_2
    else
        if deg d_2 < deg d_1 then
            Scegliere un altro primo p_1
        else
            d_2 = u d_2 ∈ 𝔽_{p_2} in notazione bilanciata
            d_1 = CRA(d_1, d_2, p_1, p_2) in notazione bilanciata
            p_1 = p_1 p_2
            d_12 = d_1 / c(d_1) ∈ ℤ[x]
            if p_1 > 2B and d_12|f and d_12|g then return d_12
            else
                Scegli p_2 e itera
            end if
        end if
```

```
    end if
end if
```

Cerchiamo ora un bound adatto. Definiamo la norma di un polinomio come

$$\left\| \sum_{i=0}^{n} a_i x^i \right\| := \sqrt{\sum_{i=0}^{n} |a_i|^2}$$

Diamo qualche risultato preliminare:

Lemma 2.3. Sia $q(x) = \sum_{k=0}^{m} c_k x^k \in \mathbb{C}[x]$ e sia $\alpha \in \mathbb{C}$. Allora

$$\|(x+\alpha)q(x)\| = \|(\bar{\alpha}x+1)q(x)\|$$

Dimostrazione. Svolgiamo il prodotto al primo membro, ponendo $c_{-1} = 0$:

$$\begin{aligned}(x+\alpha)q(x) &= xq(x) + \alpha q(x) \\ &= \sum c_k x^{k+1} + \sum \alpha c_k x^k \\ &= c_m x^{m+1} + \sum (c_k + \alpha c_{k+1}) x^{k+1}\end{aligned}$$

Calcoliamo ora la norma:

$$\begin{aligned}\|(x+\alpha)q(x)\|^2 &= c_m \bar{c}_m + \sum (c_k + \alpha c_{k+1})(\bar{c}_k + \bar{\alpha}\bar{c}_{k+1}) \\ &= \|q\|^2 + |\alpha|^2 \|q\|^2 + \sum \bar{\alpha} c_k \bar{c}_{k+1} + \alpha c_{k+1} \bar{c}_k\end{aligned}$$

Calcoliamo ora il prodotto al secondo membro:

$$\begin{aligned}(\bar{\alpha}x+1)q(x) &= \bar{\alpha}xq(x) + q(x) \\ &= \sum \bar{\alpha} c_k x^{k+1} + \sum c_k x^k \\ &= \bar{\alpha} c_m x^{m+1} + \sum (\bar{\alpha} c_{k-1} + c_k) x^k\end{aligned}$$

La norma è quindi

$$\begin{aligned}\|(\bar{\alpha}x+1)q(x)\|^2 &= |\alpha|^2 c_m \bar{c}_m + \sum (\bar{\alpha} c_{k-1} + c_k)(\alpha \bar{c}_{k-1} + \bar{c}_k) \\ &= \|q\|^2 + |\alpha|^2 \|q\|^2 + \sum \bar{\alpha} c_k \bar{c}_{k+1} + \alpha c_{k+1} \bar{c}_k\end{aligned}$$

da cui l'uguaglianza cercata. □

Proposizione 2.4 (Disuguaglianza di Landau)**.** Sia $p = \sum_{i=0}^{d} a_i x^i \in \mathbb{C}[x]$ e siano $\alpha_1 \ldots, \alpha_d$ radici di p. Sia

$$M(p) = |a_d| \prod_{i=1}^{d} \max(1, |\alpha_i|)$$

Allora $M(p) \leq \|p\|$

Dimostrazione. Siano $\alpha_1, \ldots, \alpha_d$ le radici di p e supponiamo che $\alpha_1, \ldots, \alpha_k$ siano le radici di p tali che $|\alpha_i| > 1$. Consideriamo allora

$$R(x) = a_d \left(\prod_{i=1}^{k} (\bar{\alpha}_i x + 1) \right) \left(\prod_{i=k+1}^{d} (x - \alpha_i) \right) = b_d x^d + \cdots + b_0$$

Per il lemma, si ha che $\|p\| = \|R\|$. D'altronde, $\|R\|^2 = \sum_{i=1}^{d} |b_i|^2 \geq |b_d|^2$. Poichè per costruzione $b_d = a_d \prod_{i=1}^{k} \bar{\alpha}_i$, si ha

$$\|p\|^2 = \|R\|^2 \geq |b_d|^2 = M(p)^2$$

da cui la tesi. □

Siamo ora pronti a dimostrare la proposizione che ci fornirà il bound sui coefficienti del gcd di due polinomi.

Teorema 2.5. Sia $p(x) = \sum_{i=0}^{n} a_i x^i$ e sia $q(x) = \sum_{i=0}^{m} b_i x^i$. Supponiamo che $q(x) \mid p(x)$. Allora

$$\sum |b_i| \leq \frac{|b_m|}{|a_n|} 2^m \|p\|$$

Dimostrazione. Siano $\alpha_1, \ldots, \alpha_m$ le radici di $q(x)$. Sappiamo allora che ogni coefficiente di q è una funzione simmetrica delle radici

$$|b_i| = |b_m| \sum \prod |\alpha_j| \leq \binom{m}{i} M(q)$$

Di conseguenza,

$$\sum_{i=0}^{m} |b_i| \leq 2^m M(q)$$

Poichè per ipotesi $q \mid p$, si ha la disuguaglianza

$$M(q) \leq \frac{|b_m|}{|a_n|} M(p) \tag{2.1}$$

In conclusione,

$$\begin{aligned} \sum_{i=0}^{m} |b_i| &\leq 2^m M(q) \\ &\leq 2^m \frac{|b_m|}{|a_n|} M(p) && \text{Per l'equazione 2.1} \\ &\leq 2^m \frac{|b_m|}{|a_n|} \|p\| && \text{per la disuguaglianza di Landau} \end{aligned}$$

□

2.2 Un algoritmo euristico per il calcolo del GCD

Illustriamo ora un algoritmo euristico per il calcolo del gcd; l'idea dell'algoritmo è quella di portare il problema su $\mathbb{Z}$ e ricostruire la soluzione in $\mathbb{Z}[x]$. Siano dunque $f, g \in \mathbb{Z}[x]$ e sia $\eta \in \mathbb{N}$. Consideriamo l'omomorfismo

$$\begin{array}{rccc} \varphi_\eta\colon & \mathbb{Z}[x] & \longrightarrow & \mathbb{Z} \\ & f(x) & \longmapsto & f(\eta) \end{array}$$

e sia $\delta = (f(\eta), g(\eta))$. Sia ora $d(x)$ un polinomio di $\mathbb{Z}$ tale che $\varphi_\eta(d) = \delta$; questa è la nostra candidata soluzione. Chiaramente esiste un modo privilegiato per scegliere $d(x)$; δ ammette infatti una scrittura in base η

$$\delta = c_0 + c_1\eta + \dots c_k\eta^k \qquad\qquad c_i < \eta \quad \forall i = 0, \dots, k$$

e dunque scegliamo $d(x) = \sum c_i x^i$. Affinché sia possibile sperare che $d(x) = (f, g)$, necessariamente η deve essere abbastanza grande in maniera che $|c_i| \leq \eta/2$. Il primo problema è capire se $d \mid (f, g)$. Ma anche se si verificasse questo, ci sarebbero ancora problemi; per esempio:

Esempio. Sia $f(x) = (x+1)(x-1)(x-9)$ e sia $g(x) = (x+1)(x-9)$. Consideriamo $\eta = 10$. Allora $f(10) = 99$ e $g(10) = 11$; di conseguenza $\delta = 11 = 1 \cdot 10^0 + 1 \cdot 10^1$. L'algoritmo, per come lo abbiamo esposto fino ad ora, restituisce quindi il polinomio $d(x) = (x + 1)$ mentre $(f, g) = (x + 1)(x - 9)$.

Cerchiamo di capire quali accorgimenti prendere in maniera tale che l'algoritmo abbia speranza di restituire una soluzione. Detto $G = (f, g)$, si ha allora che $\varphi_\eta(G) \mid \delta$.

Proposizione 2.6 (Disuguaglianza di Cauchy)**.** Sia $p(x) = \sum a_i x^i \in \mathbb{C}[x]$ e sia $\alpha \in \mathbb{C}$ una radice di p. Allora

$$|\alpha| < 1 + \frac{\max |a_i|}{|a_n|}$$

Dimostrazione. Consideriamo la matrice compagna del polinomio:

$$\begin{pmatrix} 0 & 0 & \dots & 0 & -\frac{a_0}{a_n} \\ 1 & & & & -\frac{a_1}{a_n} \\ & 1 & & & -\frac{a_2}{a_n} \\ & & \ddots & & \vdots \\ & & & 1 & -\frac{a_{n-1}}{a_n} \end{pmatrix}$$

Per il primo teorema di Gershgoring, ogni radice del polinomio ha norma minore della norma infinito della matrice, e quindi

$$|\alpha| \leq 1 + \frac{\max |a_i|}{|a_n|}$$

Se $a_0 \neq 0$ la matrice è irriducibile, dunque per il terzo teorema di Gershgoring, l'uguaglianza non può valere. Se $a_0 = 0$ si ripete il tutto con $p(x)/x$ e si ottiene comunque la tesi. □

Definiamo la norma infinito di un polinomio come il coefficiente di valore assoluto maggiore.

Proposizione 2.7. Siano $f, g \in \mathbb{Z}[x]$ e sia $N \in \mathbb{Z}$ tale che

$$|N| \geq 2 + \min\left\{\frac{\|f\|_\infty}{|a_n|}, \frac{\|g\|_\infty}{|b_m|}\right\}$$

Sia $\delta = (f(N), g(N))$ e sia $d(x)$ un polinomio per cui $d(N) = \delta$ e $d \mid f \ \wedge \ d \mid g$. Allora $d = (f, g)$.

Dimostrazione. Sia $G = (f, g)$. Per ipotesi, $d \mid G$ e dunque $G = d(x)h(x)$. Valutando in N, $G(N) = d(N)h(N)$. Poichè

$$G(N) \mid f(N) \qquad\qquad G(N) \mid g(N)$$

per definizione di gcd si ha che $G(N) \mid d(N)$, dunque $G(N) = G(N)q(N)h(N)$. Di conseguenza, $q(N)h(N) \in \mathbb{Z}^*$ e quindi $h(N) \in \mathbb{Z}^*$.

Mostriamo ora che allora anche $h(x) \in \mathbb{Z}[x]^*$. Supponiamo per assurdo che h non sia invertibile: h è allora un polinomio di grado positivo

$$h(x) = c\prod_{i=1}^{k}(x - \alpha_i)$$

La valutazione di h in N $h(N) = c\prod(N - \alpha_i)$ e poichè questo è invertibile, il suo modulo deve essere uguale a 1. In particolare, deve esistere una α_i tale che $|N - \alpha| \leq 1$. Abbiamo però che α è una radice sia di f che di g, dunque

$$\begin{aligned}
|N - \alpha| &\geq |N| - |\alpha| \\
&\geq 2 + \min\left\{\frac{\|f\|_\infty}{|a_n|}, \frac{\|g\|_\infty}{|b_m|}\right\} - |\alpha| \\
&> 1 \qquad\qquad \textit{Per la disuguaglianza di Cauchy}
\end{aligned}$$

Abbiamo quindi ottenuto un assurdo, poichè $1 < |N - \alpha| \leq 1$ e quindi h è una costante invertibile e $d = G$, come voluto. □

Questa proposizione conforta rispetto alla possibilità che l'algoritmo possa restituire in output il gcd di f e g. Rimangono ancora alcuni problemi:

Esempio. Sia $f = x(x-1)(x-2)$ e $g = (x+1)(x+2)(x+3)$. Allora, indipendentemente dalla scelta di η, le valutazioni di f, g avranno sempre un fattore 2 in comune. Infatti, tre numeri consecutivi sono sempre divisibili per 2.

Per ovviare al problema dell'esempio, si considerano in input polinomi primitivi: in questo modo il gcd dovrà essere primitivo (per il lemma di Gauss) e dunque, detto c il contenuto di d, basterà considerare in output $d'(x) = d/c$. In questo modo cambia l'enunciato della proposizione precedente:

Proposizione 2.8. Siano $f, g \in \mathbb{Z}[x]$ polinomi primitivi e sia $N \in \mathbb{Z}$ tale che

$$|N| \geq 2 + 2\min\left\{\frac{\|f\|_\infty}{|a_n|}, \frac{\|g\|_\infty}{|b_m|}\right\}$$

Sia $\delta = (f(N), g(N))$ e sia $d(x)$ il risultato dato dall'algoritmo euristico. Supponiamo che $d' \mid f \ \wedge \ d' \mid g$. Allora $d' = (f, g)$.

2.3 Fattorizzazione Squarefree

Iniziamo ora a elaborare un algoritmo per la fattorizzazione dei polinomi in un anello euclideo $K[x]$. Sappiamo che $K[x]$ è un UFD, dunque dato $f \in \mathbb{Z}[x]$ possiamo scrivere

$$f = f_1^{\alpha_1} f_2^{\alpha_2} \dots f_k^{\alpha_k}$$

dove ogni f_i è irriducibile e $(f_i, f_j) = 1$ se $i \neq j$. Chiamiamo P_i il prodotto dei fattori irriducibili f_j per i quali $\alpha_j = i$. Possiamo allora scrivere $f = P_1 P_2^2 \dots P_s^s$. Chiaramente $(P_i, P_j) = 1$ se $i \neq j$. Supponiamo ora che $\operatorname{char} K = 0$. Deriviamo f:

$$f' = \sum_{i=1}^{s} i P_i' P^{i-1} \prod_{\substack{j=1 \\ j \neq i}}^{s} P_j^j$$

Calcoliamo (f, f'). Se infatti fosse $(f, f') = P_2 P_3^2 \dots P_s^{s-1}$, avremmo ottenuto $f/(f, f') = P_1 P_2 \dots P_s$ e dunque la decomposizione squarefree del polinomio.

$$\begin{aligned}
(f, f') =& \left(P_1 P_2^2 \dots P_s^s, \sum i P_i' P_i^{i-1} \prod_{j \neq i} P_j^j \right) \\
=& P_2 P_3^2 \dots P_s^{s-1} \left(P_1 P_2 \dots P_s, \sum i P_i' \prod_{j \neq i} P_j \right) \\
=& P_2 P_3^2 \dots P_s^{s-1} \prod_{k=1}^{s} \left(P_k, \sum_i i P_i' \prod_{i \neq j} P_j \right) \\
=& P_2 P_3^2 \dots P_s^{s-1} \prod_{k=1}^{s} \left(P_k, k P_k' \prod_{j \neq k} P_j \right) \\
=& P_2 P_3^2 \dots P_s^{s-1} \prod_{k=1}^{s} \left(P_k, k P_k' \right) \\
=& P_2 P_3^2 \dots P_s^{s-1} \qquad \textit{I polinomi irriducibili non hanno radici doppie e } k \neq 0
\end{aligned}$$

Possiamo allora fornire l'algoritmo nel caso di un campo a caratteristica 0. In input supponiamo di avere un polinomio $f \in K[x]$. In output forniamo la lista $Ris = [(P_i, i)]$ dove P_i è un prodotto di irriducibili distinti, $(P_i, P_j) = 1$ e $f = \prod P_i^i$.

$Ris = [\,]$
$i = 1$
$c_1 = (f, f')$
$b_1 = f/c_1$
while $b_i \neq 1$ **do**
  $b_{i+1} = (c_i, b_i)$
  $c_{i+1} = c_i / b_{i+1}$
  $Ris = [(b_i/b_{i+1}, i), Ris]$
  $i = i + 1$
end while
return Ris

Se il campo K è a caratteristica positiva, sorgono però alcuni problemi. Può infatti succedere che la derivata di un polinomio f di grado positivo sia costantemente nulla; un esempio è il polinomio $f = x^p - 1$ su $\mathbb{F}_p$. Notiamo che se ciò accade, ogni monomio di f deve avere come esponente un multiplo di p. Supponiamo che il campo considerato sia perfetto, cioè che ogni elemento ammetta radice p-esima. Allora, ogni coefficiente a_i del polinomio ammette radice p-esima b_i e dunque

$$f(x) = \sum a_i x^{pj_i} = \left(\sum b_i x^{j_i} \right)^p = g(x)^p$$

Nel caso di un campo a caratteristica positiva, dunque, è necessario modificare l'algoritmo. Se $f = P_1 \dots P_s^s Q$, con $Q' = 0$, allora Ris sarà comunque l'array dei P_i, ma a differenza di prima, c_i non si riduce a 1, bensì resta pari a Q. Si trova dunque Q_1 tale che $Q_1^p = Q$ e si ripete l'algoritmo, ottenendo $Q_1 = R_1 R_2^2 \dots R_t^t Q_2$ con $Q_2' = 0$. Si va avanti finché c_i non sia 1, e l'algoritmo termina in quanto il grado dei c_i diminuisce sempre. Si restituisce quindi

$$f(x) = P_1 P_2^2 \dots P_s^s (R_1^p R_2^{2p} \dots R_t^{pt}(\dots))$$

.

2.4 Fattorizzazione di polinomi su campi finiti

Occupiamoci ora della fattorizzazione di polinomi a coefficienti in $\mathbb{F}_p$. Questo è il passo fondamentale verso la fattorizzazione di polinomi a coefficienti interi: in quel caso ci ridurremo come sempre a lavorare modulo un opportuno primo e cercheremo di ricavare la fattorizzazione in $\mathbb{Z}[x]$ a partire da quella ottenuta nel quoziente. Sia allora $f \in \mathbb{F}_p[x]$ un polinomio; possiamo chiaramente supporlo monico a meno di invertire il coefficiente di testa. In più, per l'algoritmo di fattorizzazione squarefree, possiamo anche supporlo squarefree. Sia n il grado di f. Chiaramente, f si scompone come prodotto di irriducibili

$$f = \prod_{i=1}^{k} f_i$$

dove $(f_i, f_j) = 1$. Per il teorema cinese del resto, vale allora

$$\mathbb{F}_p[x]/(f) \sim \prod_{i=1}^{k} \mathbb{F}_p[x]/(f_i)$$

e dunque il quoziente è un prodotto di campi finiti. Dato $g \in \mathbb{F}_p[x]/(f)$, possiamo allora vederlo come un elemento del prodotto

$$g = (g_1, \dots, g_k) \in \prod_{i=1}^{k} \mathbb{F}_p[x]/(f_i)$$

Supponiamo che $g_1 = 0$ e che $g_i \neq 0$ per ogni indice $i \geq 2$. Allora $(g, f) = f_1$; possiamo così dividere f per f_1 e iterare. L'idea dell'algoritmo è quella di trovare

degli opportuni g_i in modo tale che $(g_i, f) = f_i$. Il problema rimane chiaramente come trovare tali g_i.

Sui campi finiti sappiamo essere ben definito l'omomorfismo di Frobenius

$$\begin{array}{rccc} \varphi_p\colon & \mathbb{F}_p^n & \longrightarrow & \mathbb{F}_p^n \\ & x & \longmapsto & x^p \end{array}$$

Nel nostro caso, l'omomorfismo di Frobenius risulta essere una applicazione lineare tra $\mathbb{F}_p$-spazi vettoriali:

$$\varphi_p\colon \quad \mathbb{F}_p[x]/(f) \longrightarrow \mathbb{F}_p[x]/(f)$$

Notiamo che se $a \in \mathbb{F}_p$, allora $\varphi_p(a) = a^p = a$, ma per calcolare esattamente il fissato di φ_p, dobbiamo notare che

$$g = (g_1, \dots, g_k) \in \prod_{i=1}^{k} \mathbb{F}_p[x]/(f_i)$$

$$\varphi_p(g) = g \iff (g_1^p, \dots, g_k^p) = (g_1, \dots, g_k) \iff g_i^p = g_i \ \forall i$$

Ma in $\mathbb{F}_{p^n} \simeq \mathbb{F}_p[x]/(f_i)$, $\mathrm{Fix}(\varphi_p) = \mathbb{F}_p$, poiché gli unici elementi che soddisfano $x^p - x$ sono esattamente $\mathbb{F}_p$. Dunque $\mathrm{Fix}(\varphi_p) = \mathbb{F}_p^k$, poiché ne prendiamo una copia da ogni componente. Di conseguenza, abbiamo trovato un modo per trovare il numero di fattori irriducibili di f: è sufficiente calcolare la dimensione del nucleo dell'omomorfismo $\varphi_p - Id$. Per questo, è sufficiente scrivere la matrice che rappresenta l'omomorfismo nella base $1, x, \dots x^{n-1}$ e applicare l'algoritmo di Gauss.

Bisogna ora escogitare un espediente per calcolare effettivamente i fattori irriducibili. Sia $g \in \mathrm{Fix}(\varphi_p)$. Possiamo vedere allora g come elemento del prodotto $g \leftrightarrow (s_1, \dots, s_k)$ dove ogni s_i appartiene a $\mathbb{F}_p$. Fissato un indice i, supponiamo che $s_i \neq s_j$ quando $i \neq j$. Allora $g - s_i$ ha solo la i-esima componente uguale a 0 e dunque vale

$$f_i \mid g - s_i \qquad\qquad f_j \nmid g - s_i \ \ \forall j \neq i$$

Quindi, $(g - s_i, f) = (g - s_i, f_i) = f_i$. Abbiamo così trovato un modo per determinare le componenti irriducibili. Ovviamente non sappiamo se esistano degli s_i unici, ma in ogni caso, questo procedimento permette di fattorizzare il polinomio, e dato che sappiamo il suo numero di componenti irriducibili, sappiamo quando fermarci.

Notiamo che l'unico caso in cui questa procedura fallisce è se g ha tutte le componenti s_i uguali, ma questo vorrebbe dire che $g \in \mathbb{F}_p$, e pertanto si ovvia a ciò prendendo g fuori da $\mathbb{F}_p$.

Prima di illustrare lo pseudocodice, osserviamo che detta M_φ la matrice che che rappresenta l'omomorfismo di Frobenius φ_p, necessariamente la prima colonna di $M_\varphi - Id$ è sempre nulla e dunque $rk(M_\varphi - \mathrm{Id}) < n$. Dunque, se il rango è $n - 1$, il polinomio è irriducibile, ed è inutile continuare con il resto dell'algoritmo.

Supponiamo quindi di avere in input un polinomio $f \in \mathbb{F}_p[x]$ monico e squarefree di grado n. In output, forniremo la lista dei fattori irriducibili di f.

Inizializzare la lista dei fattori irriducibili $Irr = [\,]$
Costruire la matrice M_φ
Calcolare $r = rk(M_\varphi - \mathrm{Id})$
if $r = n - 1$ **then return** $[f]$
else
  Calcolare una base di $\mathrm{Fix}(\varphi_p)$ $\{v_1 = 1, \ldots, v_k\}$
  Scegliere $v \in \mathrm{Fix}(\varphi_p) - \mathbb{F}_p$
  for $s = 0, \ldots, p - 1$ **do**
    $g = (f, v - s)$
    if $(g \neq 1) \wedge (f \neq g)$ **then**
      $Irr = [g, Irr]$
      $f = \frac{f}{g}$
    end if
  end for
end if
if $\#Irr = k$ **then**
  return Irr
else
  Ripeti l'algoritmo su ogni fattore
end if

Poniamo che p sia un primo dispari, e consideriamo un elemento g nel fissato di φ_p. Avremo che $g^p - g \equiv 0 \pmod f$, ma allora

$$f \mid g^p - g = g(g^{\frac{p-1}{2}} - 1)(g^{\frac{p-1}{2}} + 1)$$

ossia ogni componente irriducibile di f divide esattamente uno dei tre fattori, e sappiamo anche dire quale: se $g = (g_1, \ldots, g_k)$, allora

$$f_i \text{ divide } \begin{cases} g & \text{se } g_i = 0 \\ g^{\frac{p-1}{2}} - 1 & \text{se } g_i^{\frac{p-1}{2}} - 1 = 0 \\ g^{\frac{p-1}{2}} + 1 & \text{se } g_i^{\frac{p-1}{2}} + 1 = 0 \end{cases}$$

In particolare, questo divide anche le classi di resto modulo p tra i residui quadratici e i non residui quadratici, ossia esattamente a metà se non consideriamo lo 0. Ciò vuol dire che possiamo provare a calcolare il gcd tra f e i tre fattori sopra per vedere se si scompone.

Se infatti poniamo f non irriducibile, e proviamo a fare $h = (f, g^{\frac{p-1}{2}} - 1)$, la probabilità che vada male è quando $h = f$ o 1. Ma questo succede solo quando tutti i g_i sono residui quadratici, o quando non lo sono, dunque la probabilità che $h \neq f$ o 1 è

$$P = 1 - \left(\frac{p-1}{2p}\right)^k - \left(\frac{p+1}{2p}\right)^k \geq \frac{4}{9}$$

Se proviamo a fare il gcd con tutti e tre i fattori, ci va male solo se tutti i g_i sono residui quadratici, non residui quadratici, o tutti zeri. La probabilità che ci vada bene sale a

$$P = 1 - 2\left(\frac{p-1}{2p}\right)^k - \left(\frac{1}{p}\right)^k \geq \frac{2}{3}$$

e provando con più g è molto probabile riuscire a fattorizzare f.

Commenti Il calcolo della matrice M_φ costa $O(pn^2)$ operazioni. Infatti, se $f = x^n + \sum a_i x^i$, si ha la relazione

$$x^n = -a_{n-1}x^{n-1} + \cdots + a_0$$

Supponiamo di aver calcolato $x^s = b_{n-1}x^{n-1} + \cdots + b_0$ modulo (f). Allora

$$x^{s+1} = b_{n-1}(-a_{n-1}x^{n-1} + \cdots + a_0) + b_{n-2}x^{n-1} + \cdots + b_0$$

e poichè dobbiamo ripetere questa operazione per np volte, il costo totale è $O(pn^2)$. Il calcolo del rango, con Gauss, è dell'ordine di $O(n^3)$: il calcolo di tutti i massimi comuni divisori è dell'ordine di $O(pkn^2)$. Di conseguenza, la complessità dell'algoritmo è $O(n^3 + pkn^2)$. Poiché di solito il primo scelto è molto maggiore di n, il secondo termine domina e dunque la complessità è maggiore di $O(n^3)$.

Fattorizzazione in Gradi Distinti Mostriamo ora un altro algoritmo per la fattorizzazione in $\mathbb{F}_p[x]$, basato sui seguenti due lemmi di carattere teorico:

Proposizione 2.9. Sia $q(x) \in \mathbb{F}_p[x]$ un polinomio irriducibile di grado d. Allora $q(x) \mid x^{p^d} - x$.

Dimostrazione. Sappiamo che $\mathbb{F}_p[x]/(q) \simeq \mathbb{F}_{p^d}$. Dato che per ogni elemento $a \in \mathbb{F}_{p^d}$ vale $a^{p^d} = a$, e che in particolare le radici di q sono contenute in questo campo, $x^{p^d} - x = 0 \pmod q$, cioè $q \mid x^{p^d} - x$. □

Proposizione 2.10. Sia $n \in \mathbb{N}$ e siano $\{f_i \mid i \in I\}$ tutti i polinomi irriducibili in $\mathbb{F}_p[x]$ di grado un divisore di n. Allora

$$x^{p^n} - x = \prod_{i \in I} f_i$$

Dimostrazione. Mostriamo prima che se q è un polinomio irriducibile di grado d e $d \mid n$, allora $q \mid x^{p^n} - x$. Sappiamo che $\mathbb{F}_p[x]/(q)$ è un campo con p^d elementi. Ogni radice di q soddisfa quindi $\alpha^{p^d} = \alpha$. Poichè $d \mid n$, abbiamo $n = dk$. Dunque $\alpha^{p^n} = \alpha^{p^{dk}} = \alpha$. Dunque α è radice di $x^{p^n} - x$ e dunque $q \mid x^{p^n} - x$.

Supponiamo ora che q sia un polinomio irriducibile di grado d e che $q \mid x^{p^n} - x$ e mostriamo che $d \mid n$. Sia α un elemento di $\mathbb{F}_{p^n}$ che è anche una radice di q. È ben definito l'omomorfismo

$$\begin{array}{rccc} \varphi\colon & \mathbb{F}_p[x] & \longrightarrow & \mathbb{F}_{p^n} \\ & f & \longmapsto & f(\alpha) \end{array}$$

Chiaramente, q è contenuto in $\mathrm{Ker}(\varphi)$; per massimalità dell'ideale (q), abbiamo una mappa iniettiva

$$\begin{array}{rccc} \tilde{\varphi}\colon & \mathbb{F}_p[x]/(q) & \longrightarrow & \mathbb{F}_{p^n} \\ & \bar{f} & \longmapsto & f(\alpha) \end{array}$$

e dunque in $\mathbb{F}_{p^d}$ si immerge $\mathbb{F}_{p^n}$. Per il teorema sul grado delle sottoestensioni, necessariamente $d \mid n$. □

Illustriamo ora lo pseudocodice di un algoritmo di fattorizzazione non in fattori irriducibili, ma in polinomi le cui componenti irriducibili abbiano lo stesso grado; supponiamo di avere in input un polinomio $f(x) \in \mathbb{F}_p[x]$ monico e squarefree di grado n.

$v(x) = f(x)$
$w(x) = x$
$d = 0$
$g_i = 1 \ \forall i$
while $(d \leq \frac{n}{2}) \wedge (v \neq 1)$ **do**
  $d = d + 1$
  $w(x) = w(x)^p \ mod \ v$
  $g_d = (w(x) - x, v(x))$
  if $g_d \neq 1$ **then**
    $v = \frac{v}{g_d}$
    $w(x) = (w(x) \ mod \ v)$
  end if
end while
return $g_1, \ldots, g_{\frac{n}{2}}$

Come anticipato, il risultato di questo algoritmo non sono polinomi irriducibili, dunque va preso solo come un algoritmo di pre-processing del polinomio. Per raggiungere il risultato finale, utilizziamo un metodo simile a quello visto sopra.

Sappiamo che g_k dividono $x^{p^k} - x$, ma se poniamo p dispari, allora questo si scompone in

$$x^{p^k} - x = x(x^{\frac{p^k-1}{2}} - 1)(x^{\frac{p^k-1}{2}} + 1)$$

dunque possiamo fare il gcd con uno dei fattori, e sperare che si scomponga. Se non funziona, possiamo sostituire $x \mapsto x + s$ con $s \in \mathbb{F}_p$, che rimescola gli elementi di $\mathbb{F}_p^k$ in maniera non banale, e riprovare. Questo metodo non assicura la scomposizione, ma le probabilità di successo sono molto alte.

2.5 Fattorizzazione di polinomi a coefficienti interi

Studiamo ora un algoritmo per la fattorizzazione di polinomi a coefficienti interi. L'idea è come sempre quella di ricondursi alla fattorizzazione su campi finiti; stavolta non useremo però il teorema cinese del resto per costruire la soluzione, ma l'insieme ampio che utilizzeremo sarà formato dalle potenze di un ideale fortunato. Utilizzeremo il seguente lemma:

Lemma 2.11 (di Hensel)**.** Sia $f \in \mathbb{Z}[x]$ un polinomio monico e sia $m \in \mathbb{Z}$. Supponiamo $f = g_1 h_1$ modulo m, con $(g_1, h_1) = 1$. Allora esistono unici g_2, h_2 tali che $(g_2, h_2) = 1$, $f = g_2 h_2$ modulo m^2 e $g_2 = g_1$, $h_2 = h_1$ modulo m.

Dimostrazione. Dimostreremo solo l'esistenza. Chiaramente, se tali g_2, h_2 esistono, dovrà valere

$$g_2 \equiv g_1 + mb \ (m^2) \qquad\qquad h_2 \equiv h_1 + mc \ (m^2)$$

Ci basta allora mostrare che esistono c, b che realizzano l'uguaglianza. Chiaramente, per ipotesi, varrà $f \equiv f_1 g_1 + mk \ (m^2)$ Poichè vorremmo che $f = g_2 h_2 \ (m^2)$, otteniamo

$$\begin{aligned} f \equiv & g_2 h_2 \ (m^2) \\ \equiv & (g_1 + mb)(h_1 + mc) \ (m^2) \\ \equiv & g_1 h_1 + g_1 mc + h_1 mb \ (m^2) \end{aligned}$$

Dunque

$$m(bh_1 + cg_1) \equiv mk \ (m^2) \Rightarrow bh_1 + cg_1 \equiv k \ (m)$$

Poiché $(h_1, g_1) = 1$, esistono b e c come richiesti per l'identità di Bezout.

Mostriamo ora che h_2, g_2 sono coprimi modulo m^2. Mostriamo cioè che esistono r_2, s_2 tali che $rg_2 + sh_2 = 1$. Per ipotesi, esistono r_1, s_1 tali che $r_1 g_1 + s_1 h_1 = 1 \ (m)$, dunque esiste z tale che $r_1 g_1 + s_1 h_1 = 1 + mz \ (m^2)$. Poniamo $r_2 = r_1 + mw$ e $s_2 = s_1 + my$; mostriamo che possiamo determinare w, y in modo da ottenere quanto voluto.

$$r_2 g_2 + s_2 h_2 \equiv 1 (m^2) \iff (r_1 + mw)(g_1 + mb) + (s_1 + my)(h_1 + mc) \equiv 1 \ (m^2)$$

Sviluppando i conti, otteniamo

$$wg_1 + yh_1 + r_1 b + s_1 c + z \equiv 0 \ (m)$$

e dunque $wg_1 + yh_1 \equiv -z - r_1 b - s_1 c$. Poiché s_1, r_1, z, b, c sono già fissati e $(g_1, h_1) = 1$, otteniamo la tesi.

□

Supponiamo di avere in input un polinomio f primitivo e squarefree. In output forniremo la fattorizzazione di f. C'è da dire però che questo algoritmo non termina sempre: per esempio il polinomio $x^4 + 1$ è irriducibile su $\mathbb{Z}$, ma è riducibile in ogni $\mathbb{F}_p$.

Scegliere p primo tale che $p \nmid lc(f)$
Calcolare il bound B sui coefficienti dei fattori irriducibili
Controllare che $p \nmid Ris(f, f')$; altimenti scegliere un altro primo
Fattorizzare il polinomio modulo p
if f è irriducibile in $\mathbb{F}_p[x]$ **then**
    f è irriducibile
else
    $f = \bar{f}_1 \dots \bar{f}_k \ mod \ p$
    Sollevare $\bar{f}_1 \dots \bar{f}_k$ modulo $p^{2r} > 2B$
    Verificare che $\bar{f}_1 \dots \bar{f}_k$ in notazione bilanciata dividono f in $\mathbb{Z}[x]$.
Altrimenti accoppiarli e ricombinarli.
end if

Lo stesso algoritmo può essere utilizzato su $\mathbb{Z}[x_1, \ldots, x_n]$. É sufficiente infatti lavorare modulo un ideale del tipo $I = (x_2 - a_2, \ldots, x_n - a_n)$; in questo modo ci siamo ricondotti al caso precedente e possiamo lavorare con Hensel sull'ideale I.

Capitolo 3

Polinomi Irriducibili

3.1 Polinomi irriducibili su un campo finito

Può sorgere spontaneo chiedersi quale sia la reale utilità di un algoritmo di fattorizzazione: dato un polinomio in $\mathbb{Z}[x]$ monico, qual è la probabilità che esso sia irriducibile? Per studiare questo abbiamo bisogno prima di contare i polinomi irriducibili in $\mathbb{F}_p$ di grado fissato. Più precisamente, chiamiamo $N_m(p)$ l'insieme dei polinomi monici irriducibili di grado m in $\mathbb{F}_p[x]$. Con lo scopo di trovare una formula chiusa introduciamo alcune risultati.

Definizione 3.1 (Funzione di Möbius)**.** $\forall\ n \geq 1$

$$\mu(n) = \begin{cases} 1 & \textit{se } n = 1 \\ 0 & \textit{se } n \textit{ ha fattori multipli} \\ (-1)^r & \textit{se } n \textit{ è prodotto di } r \textit{ primi distinti} \end{cases}$$

La funzione è moltiplicativa, cioè se $(m, n) = 1$, $\mu(mn) = \mu(m)\mu(n)$. Inoltre

Proposizione 3.2. Sia $n \in \mathbb{N}$ un naturale maggiore di 1. Allora $\sum_{d|n} \mu(d) = 0$

Dimostrazione. Scriviamo $n = p^e q$ con $(p, q) = 1$, e p primo. Allora

$$\begin{aligned} \sum_{d|n} \mu(d) &= \sum_{r=0}^{e} \sum_{b|q} \mu(p^r b) \\ &= \sum_{r=0}^{e} \sum_{b|q} \mu(p^r)\mu(b) && \text{Moltiplicatività di } \mu \\ &= \sum_{b|q} \mu(1)\mu(b) + \sum_{b|q} \mu(p)\mu(b) && \mu(p^r) = 0\ \forall r \geq 2 \\ &= \sum_{b|q} \mu(b) - \sum_{b|q} \mu(b) \\ &= 0 \end{aligned}$$

Notiamo che questo è vero anche per $q = 1$, e visto che ogni $n > 1$ ha almeno un fattore primo, la tesi segue. □

Utilizzando questa proprietà possiamo dimostrare una formula che ci permette trovare espressioni più convenienti per le funzioni sui naturali.

Lemma 3.3 (Formula di inversione di Möbius)**.** Sia G un gruppo e sia $f : \mathbb{N} \longrightarrow G$ una funzione.

$$F(n) \coloneqq \sum_{d|n} f(d) \quad \Longrightarrow \quad f(n) = \sum_{d|n} \mu\left(\frac{n}{d}\right) F(d) = \sum_{e|n} \mu(e) F\left(\frac{n}{e}\right)$$

Dimostrazione.

$$\begin{aligned}
\sum_{e|n} \mu(e) F\left(\frac{n}{e}\right) &= \sum_{e|n} \mu(e) \left(\sum_{d|\frac{n}{e}} f(d) \right) \\
&= \sum_{e|n} \sum_{d|\frac{n}{e}} \mu(e) f(d) \\
&= \sum_{d|n} \left(\sum_{e|\frac{n}{d}} \mu(e) \right) f(d) \qquad d \mid \frac{n}{e} \Longrightarrow de \mid n \Longrightarrow e \mid \frac{n}{d} \\
&= f(n) \qquad \text{Per il lemma 3.3, si annulla ogni termine eccetto l'}n\text{-esimo}
\end{aligned}$$

□

Teorema 3.4.

$$N_n(p) = \frac{1}{n} \sum_{d|n} \mu\left(\frac{n}{d}\right) p^d$$

Dimostrazione. Sappiamo che $x^{p^n} - x = \prod_{\deg(f_j)|n} f_j$ polinomi irriducibili, e dunque vale la seguente uguaglianza sui gradi

$$p^n = \sum_{d|n} d \; N_d(p)$$

Usando la formula di inversione di Möbius con $F(n) = p^n$ e $f(d) = dN_d(p)$ si ottiene

$$f(n) = n \; N_n(p) = \sum_{d|n} \mu\left(\frac{n}{d}\right) p^d$$

che è proprio la tesi.

□

Esempio. Se $n = 4$ $\frac{p^4 - p^2}{4}$, con $p = 4$ otteniamo 588.

3.2 Polinomi irriducibili sugli interi

Ottenuta la caratterizzazione per i campi a caratteristica un primo, vorremmo dimostrare che la maggior parte dei polinomi monici in $\mathbb{Z}[x]$ sono irriducibili. Preso un numero naturale M indicheremo con $I_n(M)$ i polinomi irriducibili contenuti in

$$P_n(M) \coloneqq \{f \in \mathbb{Z}[x] \mid (-M < a_k \leq M \; \forall k) \wedge (deg(f) = n) \wedge (f \text{ monico})\}$$

L'idea è quella di calcolare il limite per M crescente del rapporto tra le cardinalità di questi due insiemi rispetto a una precisa successione.

Lemma 3.5.

$$N_n(p) > \frac{p^n}{2n} \qquad \forall\, n > 2$$

Dimostrazione.

$$\begin{aligned} nN_n(p) &= p^n + \sum_{\substack{d|n \\ d<n}} \mu\left(\frac{n}{d}\right) p^d \\ &\geq p^n - \sum_{\substack{d|n \\ d<n}} p^d \end{aligned}$$

Se $d \mid n$ e $d < n$ allora $d \leq \frac{n}{2}$ e quindi [1]

$$\sum_{\substack{d|n \\ d<n}} p^d \leq \sum_{\substack{d|n \\ d\leq \frac{n}{2}}} p^d < p^{\lfloor \frac{n}{2} \rfloor + 1}$$

Ma

$$n \geq 3 \implies \frac{n}{2} \leq n - \frac{3}{2} \implies \left\lfloor \frac{n}{2} \right\rfloor \leq n-2 \implies \left\lfloor \frac{n}{2} \right\rfloor + 1 \leq n-1$$

dunque si ottiene

$$\begin{aligned} N_n(p) &> \frac{1}{n}(p^n - p^{\lfloor \frac{n}{2} \rfloor + 1}) \\ &\geq \frac{p^n}{n}\left(1 - \frac{1}{p}\right) \\ &\geq \frac{p^n}{2n} \end{aligned}$$

□

Il seguente teorema ci dice che fissato un grado n, quasi tutti i polinomi di grado n monici sono irriducibili.

Teorema 3.6. Siano $n \geq 2$ e $g \geq 1$. Sia M_g prodotto dei primi g primi dispari.

$$\lim_{g\to\infty} \frac{\#I_n(M_g)}{\#P_n(M_g)} = 1$$

.

Dimostrazione. Chiaramente, la cardinalità $\#P_n(M_g) = (2M_g)^n$. Stimiamo allora il numero di polinomi irriducibili in $P_n(M_g)$.

I coefficienti dei polinomi presi in considerazione sono compresi tra $-M_g$ e M_g. Passando a quoziente su $\mathbb{Z}/2M_g\mathbb{Z}[x]$ e scegliendo la rappresentazione

[1] in realtà, se poniamo p dispari, possiamo imporre

$$\sum_{d|n} p^d \leq \sum_{j<n} p^j = \frac{p^n - 1}{p-1}$$

e notare che il lemma vale per ogni n naturale.

bilanciata, i polinomi rimangono sostanzialmente invariati. Per il teorema cinese del resto abbiamo

$$\mathbb{Z}/2M_g\mathbb{Z}[x] \simeq \mathbb{F}_2[x] \times \mathbb{F}_3[x] \times \cdots \times \mathbb{F}_{p_g}[x]$$

Ogni polinomio è dunque rappresentabile tramite il vettore delle sue classi di resto in maniera univoca. Poichè il passaggio al quoziente è un omomorfismo di anelli surgettivo, se un polinomio è irriducibile nel quoziente, sarà irriducibile anche in $\mathbb{Z}[x]$. Possiamo allora contare il numero delle $(g+1)$-uple che corrispondono a polinomi irriducibili; per questo, consideriamo una singola componente del prodotto. Il lemma precedente fornisce una stima dei polinomi irriducibili in $\mathbb{F}_p[x]$

$$N_n(p) > \frac{p^n}{2n}$$

I polinomi riducibili sono allora

$$p^n - N_n(p) < p^n - \frac{p^n}{2n} = p^n\left(1 - \frac{1}{2n}\right)$$

Considerando tutte le componenti, il numero dei polinomi riducibili risulta minore di

$$2^n\left(1 - \frac{1}{2n}\right)3^n\left(1 - \frac{1}{2n}\right)\dots p_g^n\left(1 - \frac{1}{2n}\right)$$

Quindi le $(g+1)$-uple di polinomi monici di grado n tali per cui esiste una componente irriducibile è almeno

$$(2M_g)^n - (2M_g)^n\left(1 - \frac{1}{2n}\right)^{g+1} = (2M_g)^n\left(1 - \left(1 - \frac{1}{2n}\right)^{g+1}\right)$$

Abbiamo perciò

$$1 \geq \frac{\#I_n(M_g)}{\#P_n(M_g)} \geq 1 - \left(1 - \frac{1}{2n}\right)^{g+1}$$

Per $g \to +\infty$, abbiamo la tesi. □

Capitolo 4

Estensioni di campi

4.1 Estensioni Semplici

Sia K un campo a caratteristica 0 e sia α un elemento algebrico su K. Sappiamo allora che α ha un polinomio minimo $p_\alpha = \sum a_i x^i$, l'unico polinomio monico e irriducibile che si annulla in α. Allora $K(\alpha) = K[x]/(p_\alpha(x))$ e una base di questo campo come K-spazio vettoriale è data da $1, x, x^2, x^3, \ldots, x^{n-1}$, dove $n = \deg(p_\alpha)$. Dato comunque $\beta \in K(\alpha)$, possiamo allora rappresentarlo in questo modo:

$$\beta = b_0 + b_1\alpha + a_2\alpha^2 + \cdots + b_{n-1}\alpha^{n-1}$$

dove ogni $b_i \in K$. Nelle implementazioni possiamo allora rappresentare un elemento $\beta \in K(\alpha)$ come un vettore $(b_0, \ldots, b_{n-1})$ o equivalentemente un polinomio.

Esiste anche un'altra rappresentazione, più comoda in alcuni casi. Consideriamo l'applicazione

$$\begin{array}{rccc} \varphi_\beta\colon & K(\alpha) & \longrightarrow & K(\alpha) \\ & \gamma & \longmapsto & \gamma\beta \end{array}$$

Questa è un omomorfismo di K-spazi vettoriali; poichè abbiamo fissato una base, tale applicazione è rappresentata in modo unico da una matrice M_β. La prima colonna conterrà i coefficienti della rappresentazione vettoriale di β; per trovare la seconda colonna, basta moltiplicare la relazione data dalla prima colonna per α

$$\begin{aligned} \alpha\beta &= \alpha(b_0 + b_1\alpha + b_2\alpha^2 + \cdots + b_{n-1}\alpha^{n-1}) \\ &= \alpha b_0 + \alpha^2 b_1 + \cdots + \alpha^n b_{n-1} \\ &= \alpha b_0 + \alpha^2 b_1 + \cdots + \alpha^{n-1} b_{n-2} + (-a_{n-1}\alpha^{n-1} - a_{n-2}\alpha_{n-2} + \cdots + a_0) \\ &= a_0 + \alpha(b_0 - a_1) + \alpha^2(b_1 - a_2) + \cdots + \alpha^{n-1}(b_{n-2} - a_{n-1}) \end{aligned}$$

Dunque, per calcolare ogni colonna della matrice, data la rappresentazione di β in forma vettoriale, è sufficiente fare n somme per ogni colonna, e quindi il passaggio dalla forma vettoriale alla forma matriciale richiede $O(n^2)$ operazioni. Passare invece dalla forma matriciale a quella vettoriale ha invece complessità costante: basta infatti prendere la prima colonna della matrice. Riportiamo ora in tabella i costi delle principali operazioni nelle due rappresentazioni riportate

	Vettore	**Matrice**
Somma	$O(n)$	$O(n^2)$
Prodotto	$O(n^2)$	$O(n^2)$
Inverso	$O(n^3)$	$O(n^3)$

Nel caso matriciale, si passa in forma vettoriale, si svolgono le operazioni e si ritorna in forma matriciale; eseguire direttamente le operazioni avrebbe infatti ordine maggiore.

4.2 Norma e Traccia

Dato p_α polinomio minimo di α, consideriamo le sue radici $\alpha_1, \dots, \alpha_n$ e supponiamo $\alpha = \alpha_1$. Abbiamo visto che dato $\beta \in K(\alpha)$, β ammette una rappresentazione $\beta = q_\beta(\alpha)$. Definiamo rispettivamente norma e traccia di β come

$$N(\beta) = \prod_{i=1}^{n} q_\beta(\alpha_i) \qquad\qquad Tr(\beta) = \sum_{i=1}^{n} q_\beta(\alpha_i)$$

Notiamo che detto F il campo di spezzamento di p_α, si ha che ogni elemento del gruppo di Galois fissa sia $N(\beta)$ che $Tr(\beta)$ e dunque questi sono elementi di K. Notiamo che la norma è moltiplicativa, cioè $N(\beta\gamma) = N(\beta)N(\gamma)$.

Possiamo estendere la definizione ai polinomi: dato $f(x, \alpha) \in K(\alpha)[x]$, definiamo

$$N(f(x,\alpha)) = \prod_{i=1}^{n} f(x, \alpha_i) \qquad\qquad Tr(f(x,\alpha)) = \sum_{i=1}^{n} f(x, \alpha_i)$$

Osservazione 4.1. Notiamo che $f(x,\alpha) \mid N(f(x,\alpha))$ e che se $g \in K[x]$, allora $N(g(x)) = g(x)^n$. Inoltre, se $\beta \in K(\alpha)$, sia $\beta = q_\beta(\alpha)$. Allora $N(\beta) = \prod_{i=1}^{n} q_\beta(\alpha_i) = \mathrm{Ris}(p_\alpha, q_\beta) \in K$.

Teorema 4.2. Sia $f(x,\alpha) \in K(\alpha)[x]$ un polinomio irriducibile. Allora la norma $N(f(x,\alpha)) = g(x)^k$, dove $g \in K[x]$ è un polinomio irriducibile, e $k \in \mathbb{N}$.

Dimostrazione. Supponiamo $N(f) = CD$ con $C, D \in K[x]$ e $(C, D) = 1$. Per definizione $N(f) = \prod f(x, \alpha_i)$; chiamiamo $f_i = f(x, \alpha_i)$. Poichè f per ipotesi è irriducibile, poiché siamo in un UFD è anche primo e dunque vale $f \mid C \vee f \mid D$. Supponiamo $f \mid C$. Sia F la chiusura normale di $K(\alpha)$; per ogni radice del polinomio minimo di α su K esiste un automorfismo di F che manda α in α_j. Ogni automorfismo può essere esteso banalmente a $K(\alpha)[x]$

$$\begin{array}{rccc} \sigma_i\colon & K(\alpha)[x] & \longrightarrow & K(\alpha_i)[x] \\ & f(x,\alpha) & \longmapsto & f(x,\alpha_i) \end{array}$$

Poiché $C \in K[x]$, C viene fissato da ognuno di questi automorfismi, mentre f_i viene mappato su f_j. Le relazioni di divisibilità devono essere conservate da ogni σ_i; poichè $f_1 = f \mid C$, allora $f_j \mid C$ per ogni j. Di conseguenza, $N(f) \mid C$ e dunque $D \in K^*$. Ciò vuol dire che $N(f)$ ha un solo fattore irriducibile, e dunque sarà la potenza di un irriducibile. □

Sia ora $\beta \in K(\alpha)$; chiaramente il polinomio minimo di β su $K(\alpha)$ è $g(x,\alpha) = x-\beta$. Detti σ_i gli automorfismi della dimostrazione, possiamo scrivere la norma $N(g(x,\alpha)) = \prod(x-\sigma_i(\beta)) \coloneqq Q$ e poichè un polinomio lineare è banalmente irriducibile, si ha che $Q(x)$ è potenza di un irriducibile di $K[x]$. Dato che $Q(\beta) = 0$, si ha che il polinomio minimo p_β di β su K divide Q e dunque

$$Q = p_\beta^k \implies p_\beta = \frac{Q}{(Q,Q')}$$

Consideriamo ora il polinomio caratteristico p_M della matrice M_β. Notiamo che il polinomio minimo della matrice e quello di β sono uguali, in quanto p_β è irriducibile, e

$$\forall \gamma \in K(\alpha) \quad p_\beta(M_\beta)\gamma = p_\beta(\beta)\gamma = 0$$

$N(g)$ e p_M hanno lo stesso grado, e sono entrambi potenza del polinomio minimo di β, dunque coincidono. Questo ci permette di calcolare in modo semplice traccia e norma di un elemento: questi coincidono infatti rispettivamente con traccia e determinante della matrice M_β. Questo è il motivo per il quale si preferisce solitamente la rappresentazione degli elementi mediante matrice.

4.3 Fattorizzazione

Sia $f(x,\alpha) \in K(\alpha)[x]$; ci poniamo il problema di fattorizzare f nell'estensione $K(\alpha)$. Questo sarà fondamentale nel calcolo del campo di spezzamento di un polinomio di $K[x]$; servirà infatti fattorizzare il polinomio nelle varie estensioni che troveremo. Sappiamo che

$$f(x,\alpha) \mid N(f(x,\alpha)) = \prod_{i=1}^{n} F_i$$

dove F_i è una fattorizzazione in irriducibili in $K[x]$; allora vale il seguente

Teorema 4.3. Se $N(f(x,\alpha))$ è libera da quadrati, allora $f(x,\alpha) = \prod_{i=1}^{n}(f(x,\alpha), F_i)$ è la fattorizzazione in irriducibili di $f(x,\alpha)$.

Dimostrazione. Siano $v_1,\dots,v_n \in K(\alpha)[x]$ gli irriducibili che dividono $f(x,\alpha)$. Allora $N(v_i(x,\alpha)) \mid N(f(x,\alpha))$ e di conseguenza, dal teorema precedente e dal fatto che la norma di f è libera da quadrati si ha

$$N(v_i) = g^k \mid \prod F_i \implies k = 1 \ \wedge \ \exists j \ \textit{t.c.} \ N(v_i) = F_j$$

Dato che g e F_j sono irriducibili, allora avremo $v \mid g = F_j$. Inoltre

$$N(f) = N\left(\prod v_i\right) = \prod N(v_i) = \prod F_i$$

Abbiamo mostrato così che ogni fattore irriducibile di f divide almeno un F_i. Se v_i dividesse più di uno dei fattori, si negherebbe l'ipotesi che $(F_i, F_j) = 1$. Inoltre, dati due diversi v_i, non possono dividere lo stesso F_i, poiché altrimenti avrebbero la stessa norma, e quindi ci sarebbero due copie di F_i. Per ragioni di grado, infine, i v_i generano tutti gli F_i. Dunque $(f(x,\alpha), F_i)$ è irriducibile. □

Dunque, abbiamo trovato un algoritmo per il calcolo della fattorizzazione; basterà infatti calcolare la fattorizzazione della norma su $K[x]$ e calcolare i *gcd* tra f e i fattori ottenuti. Bisogna però mostrare che è sempre possibile supporre che la fattorizzazione della norma sia squarefree. Per questo, mostriamo ora che esistono solo un numero finito di elementi $s \in K$ tali che $N(f(x + s\alpha))$ non sia libera da quadrati. Ricordiamo che K ha caratteristica 0.

Teorema 4.4. Sia $f \in K[x]$ un polinomio libero da quadrati. Allora esistono un numero finito di $s \in K$ tali che $N(f(x - s\alpha))$ non sia libero da quadrati.

Dimostrazione. Siano $\beta_1, \ldots, \beta_m$ le radici distinte di f. Le radici di una sua traslazione $f(x - s\alpha)$ sono $\beta_1 + s\alpha, \ldots, \beta_m + s\alpha$. Dunque le radici di

$$N(f(x - s\alpha)) = \prod f(x - s\alpha_j)$$

sono del tipo $\beta_i + s\alpha_j$. Se la norma non fosse libera da quadrati, avremmo $\beta_i + s\alpha_j = \beta_k + s\alpha_l$. Sicuramente $l \neq j$ perchè per ipotesi le β_i sono distinte. Di conseguenza,

$$s = \frac{\beta_i - \beta_k}{\alpha_l - \alpha_j}$$

e dunque $Nf((x - s\alpha))$ non è squarefree per un numero finite di scelte di s. □

Lemma 4.5. Sia $f(x, \alpha) \in K(\alpha)[x]$ un polinomio libero da quadrati. Allora esiste $g \in K[x]$ libero da quadrati tale che $f \mid g$.

Dimostrazione. Consideriamo la decomposizione in fattori liberi da quadrati della norma $N(f(x, \alpha))$, cioè $N(f(x, \alpha)) = \prod G_i^i$. Chiaramente, $f \mid \prod G_i^i$ ma f è libero da quadrati, dunque $f \mid \prod G_i = g$. □

Proposizione 4.6. Sia $f \in K(\alpha)[x]$ un polinomio libero da quadrati. Allora esistono solo un numero finito di $s \in K$ tali che $N(f(x - s\alpha, \alpha))$ non sia libero da quadrati.

Dimostrazione. Per il lemma, esiste $g \in K[x]$ tale che $f \mid g$ e g è libero da quadrati. Di conseguenza,

$$f(x - s\alpha) \mid g(x - s\alpha) \Rightarrow N(f(x - s\alpha)) \mid N(g(x - s\alpha))$$

ma quest'ultima non è libera da quadrati solo per un numero finito di valori. □

Diamo ora lo pseudocodice di un algoritmo per il calcolo della fattorizzazione.

NormSq L'algoritmo prende in input un polinomio $f \in K(\alpha)[x]$ libero da quadrati e il polinomio minimo p_α di α e restituisce $s \in K$, $g(x, \alpha) \coloneqq f(x - s\alpha)$ e $R(x) = N(g(x, \alpha))$ libera da quadrati.

$s = 0$
$g(x, \alpha) = f(x, \alpha)$
$check = false$
while $check == false$ **do**
    $R(x) = \mathrm{Ris}_y(g(x, y), p_\alpha(y))$ $(= N(g))$

```
    if deg(R, R')! = 0 then
        check = true
    else
        s = s + 1
        g(x, α) = g(x − α, α)
    end if
end while
return (s, g, R)
```

Fattorizzazione Supponiamo di avere in input $f \in K(\alpha)[x]$ libero da quadrati. L'algoritmo restituisce in output la lista dei fattori irriducibili di f su $K(\alpha)$. L'algoritmo **fatt** è l'algoritmo di fattorizzazione di polinomi in $K[x]$.

```
(s, g, R) = NormSq(f(x, α))
lfatt = [ ]
l = fatt(R) in K[x]
if Length(l) = 1 then
    lfatt = [f]
else
    for i = 1; i ≤ Lenght(l); i = i + 1 do
        h_i(x, α) = (g(x, α), l[i]) in K(α)
        g(x, α) = g/h_i in K(α)
        h_i(x, α) = h_i(x + sα, α)
        lfatt = [h_i, lfatt]
    end for
end if
return lfatt
```

4.4 Teorema dell'elemento primitivo

Ci poniamo ora il problema di determinare il campo di spezzamento di un polinomio in $K[x]$, nel caso in cui la caratteristica del campo sia 0. Il problema è quello di estendere il campo di spezzamento che si trova dopo aver esteso K con una radice, in quanto siamo in grado di lavorare solo su estensioni semplici del campo. Utilizzeremo allora il teorema dell'elemento primitivo:

Teorema 4.7 (dell'elemento primitivo)**.** Sia K un campo $\operatorname{char} K = 0$ e siano α, β elementi algebrici su K. Allora esiste $\gamma \in K(\alpha, \beta)$ tali che $K(\alpha, \beta) = K(\gamma)$.

Il problema è realizzare questo teorema dal punto di vista algoritmico. Dimostriamo un po' di risultati che ci porteranno alla dimostrazione:

Proposizione 4.8. Sia K un campo, sia α algebrico su K e consideriamo l'estensione $K(\alpha)$, con p_α polinomio minimo di α. Sia β algebrico su K e sia $Q_\beta(x, \alpha) \in K(\alpha)[x]$ tale che $Q(\beta, \alpha) = 0$. Supponiamo che $N(Q_\beta(x, \alpha))$ sia libera da quadrati. Allora $\alpha \in K(\beta)$, e in $K(\beta)$ si ha

$$(p_\alpha(x), Q_\beta(\beta, x)) = x - \alpha$$

Dimostrazione. Siano $\alpha = \alpha_1, \alpha_2, \ldots, \alpha_n$ le radici di p_α. Per definizione, la norma $N(Q_\beta(x, \alpha_i)) = \prod_{i=1}^{n} Q_\beta(x, \alpha_i)$. Di conseguenza, α è radice sia di p_α che $Q_\beta(\beta, x)$, ma α_i non è radice di $Q_\beta(\beta, x)$ per $i \geq 2$. In caso contrario, infatti, avremmo $Q(\beta, \alpha) = Q(\beta, \alpha_i) = 0$ e dunque la norma avrebbe radici multiple, contro le ipotesi. Di conseguenza, $x - \alpha = (p_\alpha, Q_\beta(\beta, x))$. □

Proposizione 4.9. Sia $Q_\beta(x, \alpha)$ il polinomio minimo di β su $K(\alpha)$. Se $N(Q_\beta(x, \alpha))$ è libero da quadrati, allora è il polinomio minimo di β su K.

Dimostrazione. Dato che $Q_\beta(x, \alpha)$ è irriducibile, allora $N(Q_\beta(x, \alpha)) = g(x)^k$, ma dato che è libero da quadrati, allora è anch'esso irriducibile. Visto che si annulla su β, allora è il suo polinomio minimo su K. □

Sia ora β algebrico su K e sia $Q_\beta(x, \alpha) \in K(\alpha)[x]$ il suo polinomio minimo su $K(\alpha)$. In generale, $N(Q_\beta(x, \alpha))$ non è libera da quadrati, ma sappiamo che esiste $s \in K$ tale che $N(Q_\beta(x - s\alpha, \alpha))$ sia libero da quadrati. In tal caso, coincide con il polinomio minimo di $\beta + s\alpha = \gamma$. Abbiamo allora, per quanto dimostrato, che $K(\gamma) = K(\alpha, \gamma)$. Notiamo però che $\beta \in K(\alpha, \gamma)$, dunque otteniamo $K(\gamma) = K(\alpha, \beta)$. Abbiamo così trovato una dimostrazione costruttiva del teorema dell'elemento primitivo. Inoltre, poichè $\gamma = \beta + s\alpha$, abbiamo sia la rappresentazione di α che quella di β.

Diamo ora lo pseudocodice dell'algoritmo per l'elemento primitivo. In input supponiamo di avere p_α il polinomio minimo di α e $Q_\beta(x, \alpha)$ il polinomio minimo di β su $K(\alpha)$. In output restituiremo $R(x)$ il polinomio minimo dell'elemento primitivo γ su K, le rappresentazioni $A(\gamma)$ e $B(\gamma)$ rispettivamente di α e β in $K(\gamma)$.

$(s, g, R) = \textbf{NormSq}(Q_\beta) \quad (g(x, \alpha) = Q_\beta(x - s\alpha, \alpha))$
$\alpha = \textbf{linsolve}(gcd(g(\gamma, x), p_\alpha(x)))$
$\beta = \gamma - s\alpha \in K(\gamma)$
return $(R(x), \gamma, \alpha, \beta)$

dove l'algoritmo **linsolve** risolve semplicemente le equazioni di primo grado.

4.5 Algoritmo Split Field

Diamo ora lo pseudocodice dell'algoritmo per trovare il campo di spezzamento. In input, supponiamo di avere un polinomio $p(x) \in K[x]$ irriducibile. In output forniremo le radici di $p(x)$ in $K(\gamma)$ e il polinomio $R(x)$ che definisce il campo di spezzamento.

$roots = [\,]$
$polys = [p(x)]$
$minpoly = p(x)$
$newminpoly = p(x)$
$index = 1$
$\beta = \gamma$ radice di $minpoly$
$polys[index] = \frac{polys[index]}{x-\beta}$ $\star$
$new\_s = 0$
$Bpoly = x - \gamma$
$roots = [\beta, roots]$
$newfactor = [\,]$
$k = 1$
for P_i in $polys$ **do**
  $(s, g, R) =$ **NormSq**$(P_i, minpoly)$
  $L =$ **fatt**(R)
  for F in L **do**
    $f(x, \gamma) = (g(x, \gamma), F)$
    if $deg(F) > deg(newminpoly)$ **then**
      $newminpoly = F$
      $index = k$
      $new\_s = s$
      $Bpoly(x, \gamma) = f(x, \gamma)$
    end if
    $g(x, \gamma) = g(x, \gamma) \diagup f(x, \gamma)$
    $f(x, \gamma) = f(x + s\gamma, \gamma)$
    if $deg(f(x, \gamma)) == 1$ **then**
      $roots =$ [**linsolve**$(f(x, \gamma)), roots]$
    else
      $newfactors = [f, newfactors]$
      $k = k + 1$
    end if
  end for
end for
new_γ radice di $newminpoly$
$\alpha =$ **linsolve**$(gcd(minpoly, Bpoly(new\_\gamma, x)))$
$\beta = new\_\gamma - new\_s\ \alpha$
sostituiamo α a γ in $roots$
if $newfactors == [\,]$ **then return** $(newminpoly, roots)$
end if
sostituiamo α a γ in $newfactors$
$polys = newfactors$
$minpoly = newminpoly$
$\gamma = new\_\gamma$
vai a $\star$

Spiegazione dell'algoritmo [1]

Significato delle variabili:

γ e new_γ: Elemento primitivo. Alla fine dell'algoritmo indicherà il campo di spezzamento $\mathbb{K}(\gamma)$

$roots$: Vettore delle radici. Saranno espresse in relazione a γ

$polys$ e $newfactor$: Polinomi ancora da fattorizzare. A mano a mano che si va avanti con le estensioni, verranno fattorizzati.

k: numeri di elementi in $newfactors$ più 1.

$index$: Indicatore. Indica quale elemento del vettore $polys$ contiene l'ultima radice del polinomio di partenza $p(x)$ aggiunta all'estensione

β: Radice di $p(x)$. É l'ultima radice aggiunta all'estensione.

α: É uguale a γ. Polinomio valutato in new_γ. Se stiamo facendo un'estensione da γ a new_γ, allora $\alpha(new\_\gamma) = \gamma$.

s e new_s: Intero. Dati new_γ, γ, β e α, allora $new\_s\ \gamma = new\_s\ \alpha = new\_\gamma - \beta$. s invece gira tra tanti, e new_s è s solo se l'elemento primitivo si aggiorna, altrimenti è zero.

$Bpoly$: Polinomio minimo di new_γ su $\mathbb{K}(\gamma)$.

L'algoritmo prende in input un polinomio irriducibile $p(x)$, e denomina come γ e β una sua radice, che mette in $roots$. Mettendosi in $\mathbb{K}(\gamma)[x]$, divide il $p(x)$ per $x - \beta$, ottenendo un polinomio $Q(x, \gamma)$ le cui radici sono gli elementi che deve aggiungere alla sua estensione per ottenere il campo di spezzamento. Mette $k = 1$ poiché è la posizione di Q in $polys$.

Preso Q, ne fa la **NormSq**, ottenendo un polinomio $g(x, \gamma) = Q(x + s\gamma, \gamma)$ la cui norma R è libera da quadrati, e quindi fattorizza R. Se $Q = \prod Q_i$ è la sua fattorizzazione, allora R sarà il prodotto di R_i, che sono norme libere da quadrati di $g_i(x, \gamma) = Q_i(x + s\gamma, \gamma)$. Notiamo che Q_i, essendo irriducibili, saranno polinomi minimi su $\mathbb{K}(\gamma)$ delle sue radici. Inoltre, per la Proposizione 4.9, R_i saranno polinomi minimi delle radici di g_i su K, e queste sono gli elementi primitivi dell'estensione di $K(\gamma)$ con la rispettiva radice di Q_i.

Preso dunque un R_i, che chiama F, ricava il g_i corrispondente, e lo chiama f. L'estensione di K con una radice di F comprende γ, dunque il grado di F è maggiore o uguale a quello di p; l'unico caso in cui è uguale, è quando tutte le radici del Q_i corrispondente sono già dentro $\mathbb{K}(\gamma)$, ma dato che Q_i sono irriducibili, allora sono in particolare lineari. In questo caso il blocco If non viene eseguito, e si ricava la radice di Q_i in relazione a γ. Infatti si ricava $Q_i = f(x + s\gamma, \gamma)$, calcola la radice e la aggiunge a $roots$.

Quando invece il grado di F è maggiore del polinomio minimo, allora una qualsiasi radice di F è un elemento primitivo per γ e una radice di Q_i. Dunque segniamo F come il nuovo polinomio minimo, e diciamo che nel vettore (inizialmente vuoto) $newfactors$ il polinomio $Q_i = f(x + s\gamma, \gamma)$ si trova nella posizione $index = k$. Infatti dopo il blocco If, dato che f non è lineare, lo aggiungiamo all'array $newfactors$, e aumentiamo k di 1. Inoltre ci segniamo s in new_s,

[1] Porca paletta, questo algoritmo!

come segno che dobbiamo cambiare elemento primitivo, e in $Bpoly$ segniamo f, che è il polinomio minimo del nuovo elemento primitivo su $\mathbb{K}(\gamma)$.

Uscendo da tutti i For, diciamo che new_γ è il nuovo elemento primitivo, e calcoliamo γ in relazione a new_γ e lo salviamo in α. Questo funziona poiché $minpoly$ è il polinomio minimo di γ su $\mathbb{K}$, $Bpoly$ di new_γ su $\mathbb{K}(\gamma)$, ed inoltre la norma di $Bpoly$ è la norma di f, cioè F, che era libero da quadrati; dunque per Proposizione 4.8, il loro gcd è esattamente la funzione voluta. Inoltre la radice di $p(x)$ corrispondente sarà $new\_\gamma - new\_s\ \alpha$, che sarà un'espressione in new_γ, che salviamo in β (che aggiungeremo a $roots$ solo se abbiamo ancora termini da fattorizzare, poiché altrimenti tutti i termini erano già lineari, e dunque β è già stato aggiunto).

Aggiorniamo tutte le radici in relazione al nuovo elemento primitivo new_γ, semplicemente sostituendo α a γ in $roots$. Se non abbiamo più termini da fattorizzare, il programma è terminato, poiché allora nell'attuale estensione abbiamo già fattorizzato tutto. Altrimenti, aggiorniamo i $newfactors$ al nuovo elemento primitivo, e rilanciamo il programma da $star$, con il nuovo elemento primitivo, il nuovo polinomio minimo, i nuovi termini da fattorizzare $polys = newfactors$, il nuovo $\gamma = new\_\gamma$ e il nuovo β.

Nel caso in cui con il γ corrente si fattorizzasse tutto a termini lineari, l'inizializzazione di $Bpoly = x - \gamma$ e $new\_s = 0$ fa sì che $\alpha = \gamma = new\_\gamma$, garantendo così l'esattezza dell'algoritmo.

Capitolo 5

Radici di Sistemi Polinomiali

5.1 Radicale di un ideale zero-dimensionale

Ci occupiamo ora del calcolo del radicale di un ideale I in un anello di polinomi. In particolare, considereremo solo particolari ideali, ma prima di definirli, abbiamo bisogno di qualche nozione:

Definizione 5.1. La *Dimensione di Krull* di un anello A è il sup delle lunghezze della catene di primi strettamente crescenti contenute in essa (può essere anche infinita). In particolare, data una catena

$$P_0 \subsetneq P_1 \subsetneq \cdots \subsetneq P_n$$

questa ha lunghezza n. Indicheremo la dimensione di Krull di A come $\dim_k(A)$.

Definizione 5.2. Dato un anello A, e un suo ideale I, allora la *Dimensione* o *Profondità* dell'ideale I è la dimensione di Krull del quoziente. In formule

$$\dim I := \dim_k \left({}^{A}\!/_{I} \right)$$

D'ora in poi, lavoreremo solo con ideali a dimensione 0 nell'anello di polinomi $K[x_1, \ldots, x_n]$, dove K è un campo algebricamente chiuso. Questo vuol dire, in particolare, che tutti i primi che contengono I sono ideali massimali. Dalla decomposizione primaria di anelli noetheriani, avremo che in realtà esistono finiti ideali massimali che contengono I, dunque il quoziente

$$K[x_1, \ldots, x_n]{}/_{\sqrt{I}} = K[x_1, \ldots, x_n]{}/_{\cap_{i=1}^{m} M_i} \cong K^m$$

ha dimensione finita come K-spazio vettoriale. Lo stesso vale per l'anello $K[x_1, \ldots, x_n]/I$, perché una K-algebra finitamente generata è prodotto finito di anelli artiniani locali che quindi hanno dimensione finita. In particolare, dunque, gli elementi $1, x_i, x_i^2, x_i^3, \ldots, x_i^m$ sono linearmente dipendenti, dove m è la dimensione del quoziente. Possiamo allora trovare un polinomio $g_i \in K[x_i]$ univariato appartenente all'ideale I, poiché corrisponderà alla combinazione degli x_i^j che si annulla nel quoziente. Basta infatti esprimere tali elementi come combinazione lineare di una base (per esempio quella dell'Escalièr), e trovare un elemento del nucleo tramite eliminazione gaussiana.

Sia allora $\tilde{g}_i$ la parte libera dai quadrati di g_i; sappiamo che $(I, \tilde{g}_1, \ldots, \tilde{g}_n) \subseteq \sqrt{I}$, e per mostrare l'uguaglianza utilizziamo per questo il seguente lemma:

Lemma 5.3. Siano J, Q ideali tali che

- Q è radicale e 0-dimensionale
- $K[x_1, \ldots, x_n] \supsetneq J \supseteq Q$

Allora $J = \sqrt{J}$.

Dimostrazione. È sufficiente ragionare per corrispondenza degli ideali sul quoziente. Dato che Q è radicale e 0-dimensionale, per il teorema cinese del resto vale

$$K[x_1, \ldots, x_n]/Q \simeq K^m$$

e dunque il quoziente è isomorfo a un prodotto di campi. Dato che gli ideali di un prodotto sono prodotto di ideali, necessariamente un ideale di K^m è radicale, in quanto deve coincidere con K^h, $h \leq m$ e tutti questi sono radicali. Per corrispondenza allora J è radicale. □

Mostriamo allora che $(I, \tilde{g}_1, \ldots, \tilde{g}_n) = \sqrt{I}$. Sappiamo che $Q = (\tilde{g}_1, \ldots, \tilde{g}_n)$ è zero dimensionale, poiché ogni primo sopra di lui contiene un termine lineare univariato in ogni variabile, e dunque è massimale. Dato che $Q \subseteq (I, \tilde{g}_1, \ldots, \tilde{g}_n)$, ci basta mostrare che Q è un ideale radicale.

Lemma 5.4. Sia $I = (f_1(x_1), \ldots, f_n(x_n)) \subseteq K[x_1, \ldots, x_n]$, con tutti gli f_i liberi da quadrati. Allora I è radicale.

Dimostrazione. Mostriamo l'enunciato per induzione sul numero di variabili. Se $n = 1$, supponiamo $f_1(x_1) = \prod x_1 - \alpha_i$. Allora, per il teorema cinese del resto

$$K[x_1]/(f_1) \simeq \prod K[x_1]/(x_1 - \alpha_1) \simeq \prod K$$

e dunque essendo un prodotto di campi è privo di nilpotenti; di conseguenza l'ideale è radicale.

Per l'ipotesi induttiva, utilizziamo il secondo teorema di omomorfismo:

$$\begin{aligned} K[x_1, \ldots, x_n]/(f_1, \ldots, f_n) &\simeq K[x_1, \ldots, x_n]/(f_1)/(f_2, \ldots, f_n) \\ &\simeq \prod K[x_2, \ldots, x_n]/(f_2, \ldots, f_n) \end{aligned}$$

e dunque anche in questo caso si tratta di un prodotto di campi.

□

Da qui concludiamo che $\sqrt{I} = (I, \tilde{g}_1, \ldots, \tilde{g}_n)$.

5.2 Sistemi polinomiali

Supponiamo di avere un sistema polinomiale Σ e di voler trovare le soluzioni di tale sistema. Premettiamo dei risultati sugli ideali 0 dimensionali:

Lemma 5.5. Dato I ideale di un anello di polinomi su un campo algebricamente chiuso, allora sono equivalenti

$$|V(I)| \text{ finita} \iff \dim_K K[x_1, \ldots, x_n]/I \text{ finita} \iff \dim I = 0$$

ed in questo caso,

$$I = \sqrt{I} \iff \dim_K {K[x_1, \ldots, x_n]}/{I} = |V(I)|$$

Se vediamo Σ come un ideale I, allora le sue soluzioni saranno $V(I)$. Sappiamo dal Teorema degli Zeri di Hilbert che $V(I)$ è non vuoto se e solo se I non è tutto l'anello di polinomi. Il lemma sopra ci dice inoltre che il numero di soluzioni è finito se e solo se I è 0-dimensionale.

Supponiamo ora che il numero di soluzioni di Σ sia m e sia G una base di Gröbner dell'ideale radicale e 0-dimensionale I rispetto all'ordinamento lessicografico. Sappiamo che nella base di Gröbner compare un polinomio g nella sola variabile x_n. Se il polinomio g avesse grado esattamente m, allora gli elementi $1, x_n, x_n^2, \ldots, x_n^{m-1}$ sarebbero una base di ${K[x]}/{I}$ e dunque tutte le altre variabili si scriverebbero come combinazione lineare di questi, cioè esisterebbero dei polinomi $p_i(x_n)$ tali che $x_i = p_i(x_n) \pmod I$. Di conseguenza, nella base di Gröbner comparirebbe il polinomio $x_i - p_i(x_n)$. e la base di Gröbner sarebbe della forma $(x_1 - p_1(x_n), \ldots, x_{n-1} - p_{n-1}(x_n), g(x_n))$. Questa forma normale rende molto semplice la risoluzione del sistema; basta infatti trovare le radici di $g(x_n)$ per poi ottenere le altre soluzioni. A livello pratico, per trovare le radici si preferiscono al calcolo simbolico metodi numerici che risultano molto più efficienti. Purtroppo, non tutti gli ideali ammettono tale forma:

Esempio. Sia $I = (x^2 - y, y^2 - 1)$. Le soluzioni del sistema sono $(\pm 1, 1), (\pm i, -1)$ e dunque sono 4, mentre il grado di $y^2 - 1$ è 2. Utilizziamo il cambio di variabili lineare

$$x_1 = x \qquad\qquad y_1 = x + y$$

Dopo aver calcolato la base di Gröbner, abbiamo

$$I \rightsquigarrow (x_1 + \frac{2}{5}y_1^3 - \frac{1}{5}y_1^2 - \frac{1}{5}y_1, y_1^4 - 2y_1^2 - 4y_1)$$

e quindi siamo riusciti a portare il sistema nella forma normale desiderata.

L'esempio suggerisce che sia sempre possibile portare il sistema in questa forma tramite una trasformazione lineare. In effetti, così è.

Proposizione 5.6. Siano $\alpha_1, \ldots, \alpha_m$ le soluzioni del sistema e consideriamo i polinomi $u_i(x) = \sum_{j=1}^{n} i^{j-1} x_j = x_1 + ix_2 + i^2 x_3 + \cdots + i^{n-1} x_n$. Consideriamo l'insieme

$$A = \{u_i \mid 0 \leq i \leq \binom{m}{2}(n-1)\}$$

Allora esiste $u_i \in A$ tale che $u_i(\alpha_k) \neq u_i(\alpha_l)\ \forall k \neq l$.

Dimostrazione. Dalla relazione $u_i(\alpha_k) - u_i(\alpha_l) = 0$ otteniamo

$$(\alpha_{k,1} - \alpha_{l,1}) + i(\alpha_{k,2} - \alpha_{l,2}) + \cdots + i^{n-1}(\alpha_{k,n} - \alpha_{l,n}) = 0$$

Di conseguenza, il polinomio $p(t) = \sum t^i(\alpha_{k,i+1} - \alpha_{l,i+1})$ è un polinomio di grado $n-1$ che si annulla in i; per motivi di grado ha al più $n-1$ radici, dunque vi sono solo $n-1$ polinomi tra gli u_i che non separano α_l e α_k. Poiché le possibili coppie sono $\binom{m}{2}$, vi sono al più $\binom{m}{2}(n-1)$ polinomi degli u_i che non separano le radici. Ma la cardinalità di A è $1 + \binom{m}{2}(n-1)$, da cui la tesi. □

Dunque, è possibile trovare un polinomio che separa tutti gli elementi di $V(I)$ come combinazione lineare delle variabili, poiché sono pochi i polinomi che non soddisfano tale condizione.

Proposizione 5.7. Siano $\alpha_1, \ldots, \alpha_m$ le radici di I. Allora è possibile costruire una famiglia $g_1, \ldots, g_m$ di polinomi tali che

$$g_i(\alpha_i) = 1 \qquad\qquad g_i(\alpha_j) = 0 \ \forall j \neq i$$

Dimostrazione. Per il teorema precedente, esiste u tale che $u(\alpha_i) \neq u(\alpha_j)\ \forall i \neq j$. I polinomi cercati sono allora

$$g_i(x) = \prod_{i \neq j} \frac{u(x) - u(\alpha_j)}{u(\alpha_i) - u(\alpha_j)}$$

□

Teorema 5.8 (Generalized Chinese Remainder Theorem)**.** Siano I_1, I_2 ideali della forma

$$I_1 = (x_1 - p_1(x_2, \ldots, x_n), J_1) \qquad I_2 = (x_1 - p_2(x_2, \ldots, x_n), J_2)$$

e supponiamo che

- $(J_1, J_2) = 1$
- $J_i = I_i \cap K[x_2, \ldots, x_n]$

Allora esiste $q \in K[x_2, \ldots, x_n]$ tale che $I_1 \cap I_2 = (x_1 - q(x_2, \ldots, x_n), J_1 J_2)$.

Dimostrazione. Per il teorema cinese del resto, esiste un unico polinomio $q(x_2, \ldots, x_n)$ tale che

$$\begin{aligned} q \equiv p_1 \pmod{J_1}, \quad q \equiv p_2 \pmod{J_2} &\implies q - p_1 \in J_1, \quad q - p_2 \in J_2 \\ &\implies x_1 - q(x_2, \ldots, x_n) \in I_1 \cap I_2 \end{aligned}$$

e dunque abbiamo

$$I_1 \cap I_2 \supseteq (x_1 - q(x_2, \ldots, x_n), J_1 J_2)$$

Sappiamo inoltre che

$$I_1 \cap I_2 \cap K[x_2, \ldots, x_n] = J_1 \cap J_2 = J_1 J_2$$

e pertanto ogni polinomio in $I_1 \cap I_2$ si riduce tramite $x_1 - q(x)$ ad un polinomio in $J_1 J_2$, mostrando l'uguaglianza. □

Definizione 5.9. Un ideale I si dice in posizione generale se

- $I = \sqrt{I}$
- Nella base di Gröbner rispetto all'ordinamento lex si ha che il polinomio $p_n(x_n)$ ha grado uguale a $\#\mathcal{V}(I)$.

Va da sè che un ideale in posizione generale sia anche di dimensione zero, poiché la sua varietà è finita. Dunque I in posizione generale è l'intersezione di un numero finito di massimali.

Dato che la base di Gröbner di un ideale massimale contenente I, anche in un campo non algebricamente chiuso, è del tipo

$$\mathfrak{M} = (x_1 - p_1(x_n), \dots, x_{n-1} - p_{n-1}(x_n), h(x_n))$$

con h irriducibile, otteniamo, applicando ripetutamente il teorema, che un ideale in posizione generale è del tipo $I = (x_1 - q_1(x_n), x_2 - q_2(x_n), \dots, h_1(x_n) \dots h_k(x_n))$, con gli h_i relativi agli ideali massimali distinti.

Teorema 5.10 (Shape-Lemma)**.** Sia I un ideale radicale 0-dimensionale. Allora per quasi tutte le trasformazioni lineari di coordinate, la base in ordine lessicografico, e ridotta, è della forma

$$I = (x_1 - p_1(x_n), \dots, x_{n-1} - p_{n-1}(x_n), p_n(x_n))$$

e I è in posizione generale.

Dimostrazione. Sia $\mathcal{V}(I) = \{\alpha_1, \dots, \alpha_m\}$. Cerchiamo $L\colon K^n \to K$ lineare tale che $L(\alpha_i) \neq L(\alpha_j)$. Sia C il vettore che rappresenta tale funzionale. Basta allora che C non appartenga all'ortogonale dei vettori $\alpha_i - \alpha_j$ al variare di i, j. Poiché vi sono $\binom{m}{2}$ scelte di i, j, abbiamo che le scelte di C che non soddisfano la tesi sono corrispondenti all'unione di $\binom{m}{2}$ iperpiani di K^n. [1]

Operando ora il cambio di coordinate in cui si fissano tutto gli x_i tranne l'ultimo, che si converte in $x_n \mapsto a_1 x_1 + \dots a_n x_n$, si ottiene che il grado del polinomio $p_n(x_n)$ deve essere pari a m, dunque I è in posizione generale, e la base ha la forma voluta. □

Ritorniamo d'ora in poi nel caso K algebricamente chiuso. Sia I un ideale 0-dimensionale; sappiamo quindi che $A = {}^{K[x]}\!/_{I}$ è uno spazio vettoriale di dimensione finita su K. Sia $f \in A$, e consideriamo l'applicazione

$$\begin{array}{rccc} L_f\colon & A & \longrightarrow & A \\ & g & \longmapsto & fg \end{array}$$

L_f è un endomorfismo e si ha $L_f + L_g = L_{f+g}$, $L_f \circ L_g = L_{fg}$. Abbiamo dunque l'applicazione

$$\begin{array}{rccc} \varphi\colon & A & \longrightarrow & \mathrm{End}(A) \\ & f & \longmapsto & L_f \end{array}$$

Tale applicazione è iniettiva, perché $L_h = L_f \iff f - h = 0 \pmod I$.

Dato che $\dim_K A < \infty$, esiste un polinomio $p(t)$ tale che $p(f) = 0$. Dunque anche $p(L_f) = 0$ e quindi $p(t)$ è multiplo del polinomio minimo di L_f, che chiamiamo h_f.

Teorema 5.11. Sia I un ideale di dimensione 0 e sia $f \in A = {}^{K[x]}\!/_{I}$. Sia M_f la matrice che rappresenta L_f e sia h_f il suo polinomio minimo. Dato $\lambda \in K$, sono equivalenti:

1. $h_f(\lambda) = 0$, cioè λ è autovalore di M_f

[1] serve che il campo K sia infinito

2. Esiste $\alpha \in \mathcal{V}(I)$ tale che $f(\alpha) = \lambda$

Dimostrazione.

(2) $\Rightarrow$ (1) Sia $\alpha \in \mathcal{V}(I)$ tale che $f(\alpha) = \lambda$. Poiché $h_f(f) \in I$ si ha che $h_f(f)(\alpha) = 0$. D'altronde, $0 = h_f(f)(\alpha) = h_f(f(\alpha)) = h_f(\lambda)$, da cui la tesi.

(1) $\Rightarrow$ (2) Dimostriamo per assurdo; supponiamo cioè che $\forall \alpha \in \mathcal{V}(I)\ f(\alpha) \neq \lambda$, dove λ è un autovalore di M_f. Detti u_i i polinomi separatori del teorema 5.7, cioè tali che $u_i(\alpha_i) = 1$ e $u_i(\alpha_j) = 0\ \forall j \neq i$, definiamo il polinomio

$$p(x) = \sum \frac{u_i(x)}{(f-\lambda)(\alpha_i)}$$

Valutandolo in α si ottiene allora

$$p(\alpha) = \frac{1}{(f-\lambda)(\alpha)} \Longrightarrow (f-\lambda)(\alpha)p(\alpha) = 1 \Longrightarrow 1 - (f-\lambda)(\alpha)p(\alpha) = 0$$

Poichè la relazione vale per ogni $\alpha \in \mathcal{V}(I)$ si ha, per il Nullstellensatz, che $1 - (f-\lambda)(x)p(x) \in \sqrt{I}$. Di conseguenza, $(1-(f-\lambda)p)^k \in I$ e dunque $1 - (f-\lambda)\tilde{g} \in I$, ossia $f - \lambda$ è invertibile in A. Ma dato g autovettore relativo a λ, allora $(f-\lambda)g = 0$, e dunque $g = 0$ da cui un assurdo.

□

Sia I un ideale radicale 0-dimensionale e sia $f \in K[x]$ tale che $f(\alpha) \neq f(\beta)$ per ogni $\alpha \neq \beta \in \mathcal{V}(I)$. Consideriamo la matrice M_f associata a L_f e sia X_λ l'autospazio relativo all'autovalore λ di M_f. Fissiamo un ordinamento $<$ e una base di Gröbner $G = (g_1, \ldots, g_k)$ dell'ideale I. Sappiamo che una base di $K[x]/I$ è data dai monomi x^d tali che $x^d \notin (lt(g_1), \ldots, lt(g_k))$[2]. Possiamo allora scrivere la base come $\mathcal{B} = \{x^{d_1}, \ldots, x^{d_m}\}$. La j-esima colonna della matrice M_f sarà formata dalle coordinate di $L_f(x^{d_j})$ rispetto alla base $\mathcal{B}$.

$$f x^{d_j} = L_f(x^{d_j}) = m_{1j}x^{d_1} + \cdots + m_{mj}x^{d_m}$$

Valutando la relazione in α, otteniamo

$$\alpha^{d_j} f(\alpha) = m_{1j}\alpha^{d_1} + \cdots + m_{mj}\alpha^{d_m}$$

In notazione vettoriale, questo equivale a

$$(\alpha^{d_1}, \ldots, \alpha^{d_m}) f(\alpha) = (\alpha^{d_1}, \ldots, \alpha^{d_m}) M_f$$

Dato che tutti gli $f(\alpha)$ sono diversi al variare di α, abbiamo determinato gli autovalori di M_f, e di conseguenza anche tutti gli autovettori sinistri, poiché la dimensione della matrice coincide con quella di A, e dunque con la cardinalità di $V(I)$.

Diamo ora lo pseudocodice di un algoritmo per la risoluzione di un sistema polinomiale. Supponiamo quindi di avere in input un ideale 0-dimensionale, e di saper trovare il radicale di I e una base di A.

[2] qui stiamo usando la notazione multinomiale $x^d = x_1^{d_1} \cdot \cdots \cdot x_n^{d_n}$

Calcolare $\sqrt{I}$
Trovare f polinomio separatore lineare
Trovare $\mathcal{B}$ base monomiale di $K[x]/\sqrt{I}$
Trovare M_f matrice che rappresenta L_f rispetto a $\mathcal{B}$
Calcolare autovalori e autovettori sinistri di M_f
Per ogni autovalore, $\lambda = f(\gamma)$, con $\gamma \in V(I)$
Preso v autovettore, si ha allora $vM_f = f(\gamma)v = c_\gamma(\gamma^{d_1}, \ldots, \gamma^{d_m})$
Supponendo che il primo vettore della base sia 1, e che gli elementi dal secondo all's-esimo siano lineari
$f(\gamma)(v_1, \ldots, v_m) = c_\gamma(1, \gamma_{i_1}, \ldots, \gamma_{i_s}, \ldots,)$ dove $\gamma = (\gamma_1, \ldots, \gamma_n)$
$c_\gamma = f(\gamma)v_1$
$\gamma_{i_1} = \frac{f(\gamma)v_2}{c_\gamma} = \frac{v_2}{v_1}$
Se x_r non è nella base, allora è in $lc(\sqrt{I})$, e dunque $\sqrt{I}$ comprende il polinomio $x_r - p_r(x_{r+1}, \ldots, x_n)$
Partendo dall'x_r con r maggiore tra quelli non presenti nella base, calcoliamo $\gamma_r = p_r(\gamma_{r+1}, \ldots, \gamma_n)$

L'algoritmo effettivamente trova le radici, ma ignora in qualche senso la molteplicità. Supponiamo che I sia un ideale 0-dimensionale, non necessariamente radicale. Allora I ammette una decomposizione primaria

$$I = \bigcap_{i=1}^{m} Q_i$$

Per il teorema cinese del resto,

$$K[x]/I \simeq \prod_{i=1}^{m} K[x]/Q_i$$

Usiamo la localizzazione per definire la molteplicità di $\alpha \in \mathcal{V}(I)$. Consideriamo il massimale $\mathfrak{M}_\alpha = \{f \mid f(\alpha) = 0\}$ e $S_\alpha = A \setminus \mathfrak{M}_\alpha$. Esiste un indice i tale che $\sqrt{Q_i} = \mathfrak{M}_\alpha$. Localizzando rispetto a $\mathfrak{M}_\alpha$, otteniamo allora

$$A_\alpha := S_\alpha^{-1} A \simeq K[x]/Q_i$$

Definiamo la molteplicità di α come $\mu_\alpha = \dim_K A_\alpha$.

Proposizione 5.12. Sia I un ideale 0-dimensionale e sia $m = \#\mathcal{V}(I)$. Allora per ogni $\alpha \in \mathcal{V}(I)$ esiste $e_\alpha \in A$ tale che

- $e_\alpha^2 = e_\alpha$
- $\sum e_\alpha = 1$
- $e_\alpha e_\beta = 0$ per $\alpha \neq \beta$
- $e_\alpha(\alpha) = 1$

Dimostrazione. Sia u un polinomio separatore, cioè tale che $u(\alpha) \neq u(\beta)$ se $\alpha \neq \beta$. Consideriamo i polinomi

$$s_\alpha(x) = \prod_{\alpha \neq \beta} \frac{u(x) - u(\beta)}{u(\alpha) - u(\beta)}$$

Fissiamo $\alpha \in \mathcal{V}(I)$. Allora $s_\alpha s_\beta(\gamma)$ si annulla su ogni $\gamma \in \mathcal{V}(I)$ per $\alpha \neq \beta$; per il Nullstellensatz, $s_\alpha s_\beta \in \sqrt{I}$ e dunque esiste r_β tale che $(s_\alpha s_\beta)^{r_\beta} = 0$. Detto $r_\alpha = \max_{\beta \neq \alpha} r_\beta$, poniamo $t_\alpha = s_\alpha^r$. Notiamo che

- $t_\alpha t_\beta = 0$
- $t_\alpha(\alpha) = 1$
- $t_\alpha(\beta) = 0$

Consideriamo ora l'ideale $J = (I, \{t_\alpha\})$. $\mathcal{V}(J) = \emptyset$, perché una qualsiasi radice deve appartenere a $\mathcal{V}(I)$ ma $t_\alpha(\alpha) = 1$. Per il Nullstellensatz, vale allora $1 \in J$ e dunque abbiamo una relazione

$$1 = f + \sum c_\alpha t_\alpha$$

Poniamo allora $e_\alpha = c_\alpha t_\alpha$; questi sono esattamente gli elementi cercati, poiché $e_\alpha e_\beta = 0$, $e_\alpha(\beta) = 0$, e modulo I abbiamo

$$\sum e_\alpha(\alpha) = e_\alpha(\alpha) = 1$$

$$e_\alpha = e_\alpha \sum e_\alpha = e_\alpha^2$$

□

Mostriamo ora che $S_\alpha^{-1}A \simeq e_\alpha A$. Notiamo che $e_\alpha A$ è un anello con identità e_α; ha allora senso mostrare l'isomorfismo mediante la proprietà universale dell'anello delle frazioni. Consideriamo allora l'omomorfismo

$$\begin{array}{rccc} \varphi\colon & A & \longrightarrow & e_\alpha A \\ & a & \longmapsto & e_\alpha a \end{array}$$

Mostriamo che per ogni $s \in S_\alpha$, $\varphi(s)$ è invertibile. Infatti $e_\alpha(x)(s(x) - s(\alpha))$ si annulla su tutto $\mathcal{V}(I)$ e dunque

$$e_\alpha \underbrace{(s(x) - s(\alpha))}_{v} \in \mathfrak{N}(A) = {}^{\sqrt{I}}\!/_{I}$$

Dunque, $w = s(\alpha) + e_\alpha v \in A^*$. Allora

$$\begin{aligned} \varphi(w) &= e_\alpha(s(\alpha) + e_\alpha v) \\ &= e_\alpha s(\alpha) + e_\alpha v \\ &= e_\alpha(s(\alpha) + v) \end{aligned}$$

Quindi $e_\alpha(s(\alpha) + v)$ è invertibile, ma $s(\alpha) + v = s(\alpha) + s(x) - s(\alpha) = s(x)$ e dunque $\varphi(s) = e_\alpha s$ è invertibile. Dunque abbiamo la mappa

$$S_\alpha^{-1}A \longrightarrow e_\alpha A$$

Mostriamo che la mappa è iniettiva. Dobbiamo cioè verificare che se $e_\alpha g = 0$ esiste $s \in S_\alpha$ tale che $sg = 0$; ma questo è banale perché $e_\alpha \in S_\alpha$. La surgettività è ovvia perché φ è surgettiva.

Teorema 5.13. Sia I un ideale 0-dimensionale, sia $f \in A$. Per ogni $\alpha \in \mathcal{V}(I)$ vale

- $L_f(A_\alpha) \subseteq A_\alpha$
- $L_f|_{A_\alpha}$ ha un unico autovalore $f(\alpha)$ di molteplicità $\mu_\alpha = \dim(A_\alpha)$

Dimostrazione. Sfruttando l'isomorfismo $A_\alpha \simeq e_\alpha A$, si ha

$$L_f(A_\alpha) = L_{fe_\alpha}(A) = e_\alpha L_f(A) \subseteq e_\alpha A$$

da cui il primo punto. Consideriamo ora $f - f(\alpha)$. Poiché $v = e_\alpha(f - f(\alpha))$ si annulla in tutte le radici di I, si ha che $v \in \sqrt{I}$ e quindi la matrice associata a $L_{e_\alpha(f-f(\alpha))}$ è nilpotente. Dunque gli autovalori sono tutti nulli; d'altronde, su A_α gli elementi sono tutti della forma $e_\alpha p$ e dunque la matrice $L_{f-f(\alpha)}$ è nilpotente. Dunque tutti gli autovalori della matrice sono $f(\alpha)$. □

Teorema 5.14 (Stickelberger)**.** Sia $f \in K[x]$ e sia I un ideale 0-dimensionale. Sia L_f l'endomorfismo di $A = {}^{K[x]}\!/_{I}$ dato dalla moltiplicazione per f. Allora

- $\mathrm{Tr}(L_f) = \sum_{\alpha \in \mathcal{V}(I)} \mu_\alpha f(\alpha)$
- $\det(L_f) = \prod_{\alpha \in \mathcal{V}(I)} f(\alpha)^{\mu_\alpha}$
- Il polinomio caratteristico è $p_{L_f}(t) = \prod_{\alpha \in \mathcal{V}(I)} (t - f(\alpha))^{\mu_\alpha}$

Usiamo le informazioni appena ricavate, e riportiamo qualche risultato senza dimostrazione. Prima di tutto, presi due polinomi $f, u \in A$, definiamo

$$g_u(f, t) = \sum_\alpha \mu_\alpha f(\alpha) \prod_{\beta \neq \alpha} (t - u(\beta))$$

dove gli α e β sono elementi di $V(I)$, con I 0-dimensionale.

Lemma 5.15.

- $g_u(f, t) \in K[t]$
- se u separa gli elementi di $V(I)$, allora

$$\beta \in V(I) \implies f(\beta) = {}^{g_u(f, u(\beta))}\!/_{g_u(1, u(\beta))}$$

Corollario 5.16 (RUR)**.** Dato $u \in A$ separatore, $p_u(t)$ il suo polinomio caratteristico (ossia della matrice M_u), e $\alpha \in V(I)$. Allora

- $u(\alpha)$ radice di $p_u(t)$ (Stickelberg)
- il numero di fattori irriducibili di $p_u(t)$ è uguale a $|V(I)|$ (Stickelberg)
- la molteplicità di α in $V(I)$ è uguale alla molteplicità di $u(a)$ come radice di $p_u(t)$
- Se y è radice di $p_u(t)$, allora

$$\left(\frac{g_u(x_1, u(y))}{g_u(1, u(y))}, \ldots, \frac{g_u(x_n, u(y))}{g_u(1, u(y))} \right)$$

  risolve il sistema di I con la stessa molteplicità di y

5.3 Forme Quadratiche

Per ultimo, citiamo qualche risultato sulle forme quadratiche, senza dimostrarli.

Definizione 5.17. Dato $f \in A$, definiamo l'applicazione bilineare $T_f : A \times A \to K$ data da $T_f(h, g) := tr(L_{fgh})$. La forma quadratica associata $Q_f(g) := tr(L_{fg^2})$ si dice **Forma Quadratica di Hermite**.

Un paio di risultati che valgono in questo caso sono

Teorema 5.18. $f \in \sqrt{I} \iff T_1(f, g) = 0 \quad \forall\, g \in A$

Teorema 5.19. $f \in A \implies rk(Q_f) = |\{\alpha \in V(I) | f(\alpha) \neq 0\}|$

In particolare, per $f = 1$, otteniamo che

$$rk(Q_1) = |V(I)| \qquad Q_1(g) = tr(L_{g^2})$$

Se il campo considerato è un sottocampo dei reali, allora possiamo chiederci se abbiamo informazioni sulle radici reali. Notiamo che in ogni caso T_f è a valori reali, e dato che è una forma simmetrica bilineare, è associata ad una matrice simmetrica reale. Ciò implica che ha tutti gli autovalori reali, e vale che

Teorema 5.20. La segnatura della matrice associata a T_f, intesa come la differenza tra autovalori positivi e negativi della matrice, è

$$|\{\alpha \in V(I) \cap \mathbb{R}^n | f(\alpha) > 0\}| - |\{\alpha \in V(I) \cap \mathbb{R}^n | f(\alpha) < 0\}|$$

e il suo rango è

$$|\{\alpha \in V(I) \cap \mathbb{R}^n | f(\alpha) \neq 0\}|$$

Capitolo 6

Basi di Gröbner su Moduli

6.1 Ordinamenti monomiali e graduazioni

Consideriamo l'anello dei polinomi in n variabili. Sappiamo che un ordinamento monomiale è una relazione d'ordine su $\mathbb{N}^n$ che rispetta le seguenti proprietà:

- è un buon ordine
- $\forall \alpha, \beta, \gamma \in \mathbb{N}^n \ \alpha < \beta \Rightarrow \alpha + \gamma < \beta + \gamma$

È possibile indurre un ordinamento monomiale tramite matrici. Sia M una matrice di rango n e siano $\alpha, \beta \in \mathbb{N}^n$. Diciamo che

$$x_1^{\alpha_1} x_2^{\alpha_2} \dots x_n^{\alpha_n} >_{\sigma_M} x_1^{\beta_1} x_2^{\beta_n} \dots x_n^{\beta_n} \iff M(\alpha - \beta) >_{lex} 0 \iff M(\alpha) >_{lex} M(\beta)$$

Tale relazione è compatibile con le operazioni, è un ordine totale perché

$$M(\alpha - \beta) \not> 0 \wedge M(\alpha - \beta) \not< 0 \implies M(\alpha - \beta) = 0 \implies \alpha - \beta = 0$$

Affinché sia un buon ordine, è necessario che su ogni colonna il primo elemento non nullo sia positivo. Infatti, se per esempio la prima colonna avesse il primo elemento non nullo negativo, avremmo

$$0 > x_1 > x_1^2 > x_1^3 > \dots$$

che è una catena discendente infinita. Questa condizione è anche sufficiente, poiché $x_i > 0$ per ogni i implica che $x^\alpha > 0$ per ogni $\alpha \in \mathbb{N}^n$ non nullo[1], e dunque dato un qualsiasi insieme di $\{x^\alpha\}$, le immagini tramite M ammetteranno un minimo in lex, che è un buon ordine su $\mathbb{N}^n$.

Esempio.

- L'ordinamento *Deglex* è indotto dalla matrice

$$\begin{pmatrix} 1 & \dots & 1 \\ & & 0 \\ & I_{n-1} & \vdots \\ & & 0 \end{pmatrix}$$

[1] usiamo x^α come abbreviazione di $x_1^{\alpha_1} x_2^{\alpha_2} \dots x_n^{\alpha_n}$, così come useremo $K[x]$ come abbreviazione di $K[x_1, x_2, \dots, x_n]$

- L'ordinamento *Degrevlex* è indotto dalla matrice

$$\begin{pmatrix} 1 & & \dots & 1 \\ 0 & & & -1 \\ \vdots & & \iddots & \\ 0 & -1 & & \end{pmatrix}$$

Teorema 6.1. Sia τ un ordinamento totale su $\mathbb{Q}^n$ e siano $u_1, \dots, u_n \in \mathbb{R}^n$ elementi linearmente indipendenti. Allora esiste $s \leq n$, ed un mappa

$$\begin{array}{rccc} \alpha\colon & (\mathbb{Q}^n, \tau) & \longrightarrow & (\mathbb{R}^s, lex) \\ & v & \longmapsto & (u_1^t \cdot v, \dots, u_s^t \cdot v) \end{array}$$

che sia ordinata e iniettiva. Detta inoltre $d(u)$ la dimensione dello spazio vettoriale su $\mathbb{Q}$ generato dalle componenti di $u \in \mathbb{R}^n$, si ha che se

$$\sum_{i=1}^{s} d(u_i) \geq n$$

allora la matrice

$$\begin{pmatrix} u_1^t \\ u_2^t \\ \vdots \\ u_s^t \end{pmatrix}$$

induce un ordine monomiale su $\mathbb{Q}^n$.

Il teorema in particolare mostra che ogni ordinamento è rappresentato da una matrice.

Osservazione 6.2. La corrispondenza matrici-ordinamenti non è biunivoca. Le matrici

$$\begin{pmatrix} \lambda_1 & 0 \\ \lambda_2 & \lambda_3 \end{pmatrix}$$

inducono, $\forall \lambda_1, \lambda_3 > 0$ e per ogni λ_2 l'ordinamento *lex*.

Definizione 6.3. Sia R un anello e siano $\{R_i\}_{i \in \mathbb{N}}$ dei sottogruppi additivi di R. R è graduato dagli R_i se

$$R = \bigoplus_{i \in \mathbb{N}} R_i$$

e $R_i \cdot R_j \subseteq R_{i+j}$.

Una graduazione di $\mathbb{K}[x_1, \dots, x_n]$ si può dare anche mediante vettori. Dato un vettore v, possiamo definire

$$R_i = \mathrm{Span}\{x^\alpha \mid v^t \cdot \alpha = i\}$$

e questo fornisce una graduazione. Notiamo che la gradazione usuale è indotta dal vettore $(1, 1, \dots, 1)$. Se il vettore v è tale che $v_i > 0$ per ogni componente, si ha che $\dim_K(R_i) < \infty$ per ogni i.

Dato che nella definizione di anello graduato importa solo che l'indicizzazione avvenga su un monoide, possiamo anche ampliare la graduazione a $\mathbb{N}^n$ allo

stesso modo. In particolare, data una matrice di naturali M anche rettangolare, possiamo definire la gradazione di x^α come $M\alpha$, che in generale sarà un vettore di naturali. Un particolare tipo di gradazioni sono le cosiddette **Gradazioni Fini**, ossia delle matrici M di rango n. La proprietà peculiare di queste gradazioni è che

Lemma 6.4. Data una gradazione fine, allora per ogni grado esiste al massimo un monomio di quel grado. Se la matrice è anche invertibile, allora ne esiste esattamente uno.

Chiaramente, dato un ordinamento e fissata una matrice che lo induce, esiste una gradazione associata, data dalla prima riga della matrice. In questo caso, la gradazione e l'ordinamento vengono detti **Compatibili**.

Una domanda che possiamo porci è se dato un polinomio in $K[x]$ multivariato, esiste una gradazione che lo renda omogeneo. In generale, se ne esiste una, allora ce n'è anche una determinata da un vettore, ma è facile vedere che per esempio $x^2 + x + y \in K[x, y]$ è omogeneo se e solo se il vettore è nullo. Dunque in generale NON esiste una tale gradazione.

6.2 Ordinamenti su Moduli

Passiamo ora ai moduli:

Definizione 6.5. Sia $F = K[x_1, \ldots, x_n]^r$. Un **monomial term ordering** su F è un ordinamento totale sui monomi compatibile con l'operazione, che sia anche un buon ordine, cioè $x_i e_j > e_j$ per ogni i, j, con e_i che indicano la componente in F.

I term ordering si dividono principalmente in due classi: quelli che su ogni copia di $K[x_1, \ldots, x_n]$ utilizzano ordinamenti diversi o quelli che utilizzano su ogni copia lo stesso ordinamento. Questi ultimi vengono detti *ordinamenti di Riquer* e faremo sempre riferimento a questi.

Tra questi, possiamo ancora distinguere 2 sottoclassi. Diremo che un ordinamento è *pos+to* se prima conta la componente di F che si prende in considerazione e poi l'ordine su $K[x_1, \ldots, x_n]$; *to+pos* viceversa.

Teorema 6.6. Sia μ un monomial term ordering di Riquet su $K[x]^r$. Allora esistono $e_1, \ldots, e_s$, esiste un ordinamento monomiale μ' su $K[x, e_1, \ldots, e_s]$ e t_i monomi di $K[x, e]$ linearmente indipendenti su $K[x]$ tali che la mappa

$$\begin{array}{ccccc} \varphi & (K[x]^r, \mu) & \longrightarrow & (K[x, e], \mu') \\ & e_i & \longmapsto & t_i \end{array}$$

sia ordinato, iniettivo e $\mu'|_{\operatorname{Im}\varphi} = \mu$.

Il teorema garantisce che possiamo vedere un ordinamento di Riquet sul modulo come un ordinamento monomiale su un anello di polinomi, ma prova solo l'esistenza di un tale ordinamento; nella pratica si scelgono solitamente due mappe:

- La prima è usare un numero di variabili e_i pari al rango del modulo, e mandare gli e_i di $K[x]^r$ negli e_i di $K[x, e]$.

- Un altro metodo è utilizzare le potenze di un'unica variabile, mandando gli e_i in e^i.
- Spesso, vorremmo trovare un ordinamento compatibile con una gradazione che renda particolari polinomi omogenei. In questo caso, si aggiunge a e una (o più) variabile ausiliaria d, in modo che $e_i \mapsto d^j e^i$.

In tutti i casi, per il calcolo di una base di Gröbner è necessario modificare leggermente l'implementazione dell'algoritmo per rispettare le operazioni permesse nel modulo.

Definizione 6.7. Sia $A = \oplus A_i$ un anello graduato. Un A-modulo M si dice graduato se esistono sottogruppi M_i tali che

- $M = \bigoplus_{i \in I} M_i$
- $A_i M_j \subseteq M_{i+j}$

Un omomorfismo $f\colon M \to N$ tra moduli graduati si dice omogeneo se $f(M_i) \subseteq N_i$.

Notiamo che se ho una gradazione di un modulo posso shiftarla, cioè porre $M'_i = M_{i+d}$; questo può influenzare quindi anche la nozione di omomorfismo omogeneo.

6.3 Basi di Gröbner

Sia M un sottomodulo di $F = K[x_1, \ldots, x_n]^r$, sia τ un ordinamento monomiale di modulo del tipo Pos+To su F e σ l'ordinamento monomiale indotto su ogni $K[x_1, \ldots, x_n]$. Il calcolo della base di Gröbner produce una matrice

$$RGB_\tau(M) = \begin{pmatrix} RGB_\sigma(I_1) & 0 & 0 & \ldots & 0 \\ * & RGB_\sigma(I_2) & 0 & \ldots & 0 \\ \vdots & & \ddots & & \end{pmatrix}$$

ed è quindi in forma triangolare inferiore.

Definizione 6.8. Se τ è un ordinamento di modulo con la proprietà che, comunque dato $v \in K[x_1, \ldots, x_n]^r$,

$$LT_\tau(v) \in 0_s \oplus K[x_1, \ldots, x_n]^{r-s} \Rightarrow v \in 0_s \oplus K[x_1, \ldots, x_n]^{r-s}$$

allora τ è detto di eliminazione per le prime s componenti del modulo.

Notiamo che questo è analogo agli ordinamenti di eliminazione per gli anelli. In quel caso, infatti, ricordiamo che un ordinamento τ si dice di eliminazione se ha la proprietà che

$$LT(f)_\tau \in K[x_s, \ldots, x_r] \Rightarrow f \in K[x_s, \ldots, x_r]$$

Sappiamo per esempio che l'ordinamento *lex* è di eliminazione. In realtà, l'unico ordinamento di simultaneamente di eliminazione per ogni variabile è lex. Per i moduli, l'analogo si ha per Pos+To:

Proposizione 6.9. L'ordinamento $\text{Pos}_{1>2>\cdots>r}$+To è di eliminazione per ogni scelta di To.

Una conseguenza di ciò è per esempio che dato M un sottomodulo di $F = K[x_1, \ldots, x_n]^r$ ordinato con τ di eliminazione per $e_1, \ldots, e_s$, si ha che

$$RGB_\tau(M) \cap (0_s \oplus (K[x_1, \ldots, x_n]^{r-s}) = RGB_\tau(M \cap (0_s \oplus K[x_1, \ldots, x_n]^{r-s}))$$

6.4 Sizigie

Definizione 6.10. Siano $f_1, \ldots, f_s$ elementi di $K[x_1, \ldots, x_n]$. Definiamo il modulo delle sizigie come

$$Syz(f_1, \ldots, f_s) = \{(a_1, \ldots, a_s) \in K[x_1, \ldots, x_n]^s \mid \sum a_i f_i = 0\}$$

Chiaramente, la definizione dipende dai polinomi scelti e non dall'ideale che essi generano; scegliendo altri generatori dell'ideale, le sizigie possono cambiare. Notiamo inoltre che se l'insieme dei polinomi f_i è finito, il modulo delle sizigie è finitamente generato. Infatti è un sottomodulo di $K[x_1, \ldots, x_n]^s$, che è noetheriano. Ci chiediamo allora come sia possibile trovarne un insieme di generatori. Per questo, copiamo il procedimento seguito nell'algoritmo di Euclide per il calcolo dei coefficienti di Bezout. Dati $f_1, \ldots, f_s$, consideriamo i vettori

$$\begin{pmatrix} f_1 & f_2 & \ldots & f_s \\ e_1 & e_2 & \ldots & e_s \end{pmatrix} \tag{6.1}$$

Calcoliamo la base di Gröbner mediante l'ordinamento Pos+To di eliminazione; otteniamo allora una matrice

$$\begin{pmatrix} RGB(f_1, \ldots, f_s) & 0 \\ * & A \end{pmatrix} \tag{6.2}$$

Mostriamo che i vettori colonna della sottomatrice A sono dei generatori per il modulo delle sizigie di $f_1, \ldots, f_s$. Per questo, è fondamentale la seguente osservazione.

> Ad ogni passo dell'algoritmo, la somma delle componenti delle colonne dalla seconda all'ultima, moltiplicate per il rispettivo polinomio f_i, fornisce l'elemento in prima posizione. Questo è vero all'inizio e l'algoritmo di riduzione di Buckberger conserva questa proprietà ad ogni passo.

Sia A_i una colonna di A. Per la proprietà enunciata, banalmente queste sono sizigie, perchè la prima componente della colonna corrispondente è nulla. Viceversa, sia v una sizigia. Mostriamo che $v \in A$. Consideriamo il vettore

$$\begin{pmatrix} 0 \\ v_1 \\ \vdots \\ v_s \end{pmatrix}$$

e riduciamo questo vettore secondo la base di Gröbner data dalla matrice 6.1. Questa infatti è una base di Gröbner rispetto all'ordinamento Pos inverso, cioè $s > s-1 > \cdots > 1$. Chiaramente la riduzione produce il vettore

$$\begin{pmatrix} 0 \\ v_1 \\ \vdots \\ v_s \end{pmatrix} \to \begin{pmatrix} \sum v_i f_i \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Dato che v è una sizigia, si ha allora che tale vettore riduce a 0, e dunque sta in M. Questo implica che riduce a 0 per una qualsiasi base di Gröbner di M; in particolare questo avviene per la base di Gr̈obner della matrice 6.2. Ma dato che la prima componente del vettore è nulla, questo significa che in realtà v riduce a 0 secondo la divisione per A, da cui la tesi.

Questo si può vedere in maniera più rapida tramite

$$\sum v_i f_i = 0 \implies \sum v_i \begin{pmatrix} f_i \\ e_i \end{pmatrix} = \begin{pmatrix} 0 \\ v \end{pmatrix} \implies v \in A$$

6.4.1 Generatori Minimali

Preso un ideale $I = (f_1, \ldots, f_s)$ di $K[x_1, \ldots, x_n]$, ci chiediamo se riusciamo a trovare un set di generatori minimale tra gli stessi f_i.

Per prima cosa notiamo che

$$f_1 \in (f_2, \ldots, f_n) \iff \exists (1, h_2, \ldots, h_s) \in Syz(f_1, \ldots, f_n)$$

Ma allora, calcoliamo una BGR delle Syzigie, e rifacciamo le BGR delle righe

$$\begin{pmatrix} \\ BGR(Syz) \\ \\ \end{pmatrix} \to \begin{pmatrix} BGR(Syz_1) \\ \vdots \\ BGR(Syz_s) \end{pmatrix}$$

avremo che le righe la cui base di Gröbner è (1) corrisponderanno a f_i che possono essere ottenuti come combinazione lineare degli altri. Dunque eliminare questi f_i ci fa rimanere con un set minimale di generatori.

Notiamo che se prendiamo I omogeneo, con gli f_i omogenei, possiamo prendere l'algoritmo sopra e spezzarlo a seconda dei gradi degli f_i. Infatti se prendiamo f_1 e f_2, e poniamo che $deg(f_1) < deg(f_2)$, e che f_1 può essere eliminato dal set dei generatori, allora $f_1 = \sum h_i f_i \implies h_2 = 0$.

Dunque dividiamo gli f_i per grado. Se d è il grado minore, allora posso applicare l'algoritmo sopra solo ai polinomi di grado d. Andando avanti così, quando arriviamo al grado s, possiamo applicare l'algoritmo solo ai polinomi di grado s e quelli di grado minore che sono già stati ridotti.
Questo metodo permette di risparmiare molto in termini computazionali.

Notiamo che se gli f_i non sono omogenei, lo stesso discorso non vale: in $(x^2, y^3 - x, y^3)$ l'unico set di generatori minimali è $(y^3 - x, y^3)$, ma x^2 è il termine col grado minore.

6.5 Operazioni sui moduli

6.5.1 Sistemi

Siano A, B matrici a coefficienti in $K[x_1, \ldots, x_n]$ e supponiamo di avere il sistema $AX = B$, dove X è una matrice. L'obiettivo che ci poniamo è di risolvere il sistema; per questo, fissiamo un ordinamento τ del tipo $Pos + To$. Calcolando la base di Gröbner del modulo, otteniamo:

$$\begin{pmatrix} B & A \\ I & 0 \\ 0 & I \end{pmatrix} \xrightarrow{RGB_\tau} \begin{pmatrix} RGB_\tau(A|B) & 0 & 0 \\ * & C & 0 \\ * & D & E \end{pmatrix}$$

Teorema 6.11. Il sistema ammette soluzioni se e solo se $C = I$. Inoltre le soluzioni sono del tipo $D + \epsilon$, dove $\epsilon \in (E)$.

Dimostrazione. Sia V una matrice tale che $AV + B = 0$. Allora

$$AV + B = 0 \iff \begin{pmatrix} B & A \end{pmatrix} \begin{pmatrix} I \\ V \end{pmatrix} = 0 \iff \begin{pmatrix} I \\ V \end{pmatrix} \in Syz(B|A) = \text{Span} \begin{pmatrix} C & 0 \\ D & E \end{pmatrix}$$

Di conseguenza, la matrice $\begin{pmatrix} I \\ V \end{pmatrix}$ riduce a 0 se divisa per una base di Gröbner delle sizigie. Questo implica allora che $C = I$ (ricordiamo che la base di Gröbner si suppone ridotta) e inoltre

$$\begin{pmatrix} I \\ V \end{pmatrix} \to \begin{pmatrix} 0 \\ V - D \end{pmatrix} \to \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

e dunque $V - D \in (E)$ da cui la tesi. □

Abbiamo quindi trovato un algoritmo per risolvere i sistemi. Però si può dire qualcosa di più: notiamo in effetti che E sono le Syzigie di A, ossia $AE = 0$, dunque si può considerare come D soluzione particolare, e E soluzione omogenea di $AX = B$. Ciò vuol dire che per stabilire se il sistema ha soluzione, e per trovarne una, basta controllare che $C = I$, e in questo caso si ci può fermare, e fornire D come soluzione.

Questo metodo è anche utile in molti contesti. Per esempio, se abbiamo una funzione $K[x]$-lineare tra due moduli $K[x]^r \to K[x]^s$, e vogliamo trovare la controimmagine di un vettore B, basta risolvere $AX = B$, con A matrice associata all'applicazione. Per fare più in fretta, si può cercare la controimmagine di una base, risolvendo $AX = I$.

6.5.2 Intersezione, somma e divisioni di sottomoduli

Siano M, N sottomoduli di un modulo libero; con un abuso di notazione, indicheremo con M anche la matrice le cui colonne sono dei generatori del modulo M e lo stesso faremo con N. Supponiamo poi che $N, M \subseteq K[x_1, \ldots, x_n]^r$. Allora, scelto τ ordinamento del tipo $Pos + To$, si ottiene

$$\begin{pmatrix} M & N \\ 0 & N \end{pmatrix} \xrightarrow{RGB_\tau} \begin{pmatrix} RGB_\tau(M + N) & 0 \\ * & RGB_\tau(M \cap N) \end{pmatrix}$$

Per calcolare invece l'ideale dato dalla divisione di due sottomoduli, notiamo che vale la formula

$$(M\colon N) = \bigcap_{i=1}^{s} (M\colon n_i)$$

dove n_i sono le colonne di N. In questo modo, ci possiamo ridurre a calcolare la divisione di un modulo per il singolo vettore. Abbiamo allora

$$\begin{pmatrix} M & v \\ 0 & 1 \end{pmatrix} \xrightarrow{RGB_\tau} \begin{pmatrix} * & 0 \\ * & RGB(M : v) \end{pmatrix}$$

La stessa cosa può essere fatta nel caso di un ideale; dato cioè $f \in K[x_1, \ldots, x_n]$, per calcolare $(M : f)$ basta calcolare una base di Gröbner

$$\begin{pmatrix} M & f \cdot I \\ 0 & I \end{pmatrix} \to \begin{pmatrix} * & 0 \\ * & RGB(M : (f)) \end{pmatrix}$$

I precedenti si possono anche combinare: se per esempio avessimo $M, N \subseteq K[x_1, \ldots, x_n]^r$ due moduli, $v \in K[x_1, \ldots, x_n]^r$, e I un ideale, e volessimo calcolare $((M \cap N) : (v)) \cap I = J$, basterebbe scrivere

$$\begin{pmatrix} M & N & 0 & 0 \\ 0 & N & v & 0 \\ 0 & 0 & 1 & I \\ 0 & 0 & 0 & I \end{pmatrix} \xrightarrow{RGB_\tau} \begin{pmatrix} RGB_\tau(M+N) & 0 & 0 & 0 \\ * & * & 0 & 0 \\ * & * & * & 0 \\ * & * & * & RGB(J) \end{pmatrix}$$

Per dimostrare che gli algoritmi enunciati sono corretti, ricorriamo ai seguenti due lemmi:

Lemma 6.12. Sia M un sottomodulo di $K[x_1, \ldots, x_n]^r$ e sia I un ideale di $K[x_1, \ldots, x_n]$. Sia v_1 un elemento di $K[x_1, \ldots, x_n]^r$ e sia v_2 un elemento di $K[x_1, \ldots, x_n]^s$. Sia M' la matrice

$$\begin{pmatrix} M & I \cdot v_1 \\ 0 & I \cdot v_2 \end{pmatrix}$$

Allora $M' \cap (0_r \cap K[x]^s) = ((M\colon v_1) \cap I)v_2$. Inoltre

$$RGB(M') = \begin{pmatrix} RGB(M + Iv_1) & 0 \\ * & RGB(((M : v_1) \cap I)v_2) \end{pmatrix}$$

Lemma 6.13. Siano M, N sottomoduli di $K[x_1, \ldots, x_n]^r$ e siano f, g polinomi di $K[x_1, \ldots, x_n]$. Allora

$$\underbrace{\begin{pmatrix} M & f \cdot N \\ 0 & g \cdot N \end{pmatrix}}_{M'} \bigcap 0_s \oplus K[x_1, \ldots, x_n]^r = g((M : (f)) \cap N)$$

Inoltre,

$$RGB(M') = \begin{pmatrix} RGB(M, fN) & 0 \\ * & RGB(g((M : (f)) \cap N)) \end{pmatrix}$$

C'è un'altra operazione che si può eseguire estraendo la base di Gröbner. Ci accorgiamo infatti che dati due ideali I, J in $K[x_1, \ldots, x_n]$, allora la sequenza di ideali $I : J^n$ è stazionaria, in quanto J^n è decrescenze, e per noetherianità finita. Dunque possiamo definire

Definizione 6.14. Dati due ideali I, J, allora il limite di $I : J^n$ è chiamato il **Saturato**, e viene indicato come $I :^\infty J$.

e l'operazione che ci permette di calcolarlo tramite le Basi è

$$I :^\infty (f) = (tI, tf - 1) \cap K[x_1, \ldots, x_n]$$

6.6 Serie di Hilbert

Torniamo all'algoritmo classico per il calcolo di una base di Gröbner.

Uno dei vantaggi di avere un algoritmo è per esempio che possiamo dire quando un certo elemento ha un inverso in un quoziente. preso infatti un ideale $I \subset K[x]$, e un elemento $f \in A = {}^{K[x]}\!/_{I}$, ci chiediamo se ha un inverso $gf = 1$ in A.

Sappiamo che in ${}^{K[x,t]}\!/_{(I, tf-1)}$ un inverso di f è t, e chiederci se esiste un g in $K[x]$ per cui $gf = 1$ equivale a chiedere se $t - g$ appartiene alla BGR di $(I, tf - 1)$. Dunque abbiamo una maniera algoritmica per scoprirlo.

Prendiamo ora un ideale I omogeneo, e incominciamo ad applicare l'algoritmo per estrarre la BGR, ma lo eseguiamo decidendo di fare gli S-polinomi prima degli elementi dei gradi più piccoli, e ridurre il più possibile.
Un modo per sapere in anticipo quando fermarsi con gli S-polinomi di un determinato grado, è per esempio conoscere quanti polinomi omogenei di grado fissato non si riducono a zero, ossia conoscere

$$\dim_K \left({}^{K[x]}\!/_{I}\right)_d = \dim_K \left({}^{K[x]}\!/_{Lt(I)}\right)_d = m_d$$

così, se arriviamo a $\binom{d+n-1}{n-1} - m_d$ termini nella base con termine di testa di grado d, sappiamo che abbiamo finito, poiché non ne possiamo aggiungere altri. Per fare ciò ci conviene prendere un ordinamento grado-compatibile.

Queste utili dimensioni m_d sono contenute nella

Definizione 6.15. Funzione di Hilbert

$$H : \mathbb{N} \to \mathbb{N} \qquad H(d) = \dim_K \left({}^{K[x]}\!/_{I}\right)_d$$

e nella

Definizione 6.16. Serie di Hilbert-Poincaré

$$HP(\lambda) := \sum_d H(d)\lambda^d$$

Per calcolare queste serie, abbiamo bisogno di alcuni lemmi

Lemma 6.17. Data $0 \to M \to N \to P \to 0$ sequenza esatta corta di $K[x]$-moduli finitamente generati. Allora la dimensione come K-spazio vettoriale è additiva, ossia

$$dim_K(N) = dim_k(M) + dim_K(P)$$

Lemma 6.18. Data $F \in K[x]$ omogenea di grado d, e M un $K[x]$-modulo finitamente generato, allora la sequenza

$$0 \to {}^{M}\!/_{(0 : (F))} \to M \to {}^{M}\!/_{FM} \to 0$$

è esatta ed omogenea se al primo modulo attuiamo uno shift di grado pari a $-d$.

Se nel lemma precedente poniamo $M = {}^{K[x]}\!/_{I}$, con I monomiale, diventa

$$0 \to {}^{K[x]}\!/_{I : (F)} \to {}^{K[x]}\!/_{I} \to {}^{K[x]}\!/_{(F, I)} \to 0$$

Quindi per calcolare la serie di ${}^{K[x]}\!/_{(F, I)}$, con $J = (F, I)$, possiamo utilizzare

$$HP_{{}^{K[x]}\!/_{J}} = HP_{{}^{K[x]}\!/_{I}} - \lambda^d \cdot HP_{{}^{K[x]}\!/_{I:(F)}}$$

Dove sia I che $I : (F)$ hanno almeno un generatore in meno rispetto a J, dunque iterando, questo procedimento porta alla soluzione. Sappiamo inoltre che

$$HP_{K[x]} = \frac{1}{(1 - \lambda)^n}$$

espresso come serie formali. Dunque in generale, avremo che la serie

$$HP_{{}^{K[x]}\!/_{I}} = \frac{p(\lambda)}{(1 - \lambda)^n}$$

con il numeratore che è un polinomio dipendente da I, che indicheremo come $< I >= p(\lambda)$. Per calcolare questo polinomio, nel caso monomiale, bastano le seguenti regole algoritmiche:

Lemma 6.19. Sia I un ideale monomiale di $K[x]$ e sia $x^\alpha \in K[x]$ un monomio. Allora

- $< 0 >= 1$
- $< 1 >= 0$
- $< x^\alpha >= 1 - \lambda^{|\alpha|}$
- $< x^\alpha I >= 1 - \lambda^{|\alpha|} + \lambda^{|\alpha|} < I >$
- $< I, x^\alpha >=< I > -\lambda^{|\alpha|} < I : x^\alpha >$

Dimostrazione.

1. Sappiamo che il numero di monomi di $K[x_1, \ldots, x_n]$, ossia la dimensione di $K[x]_d$, è dato da

   $$\dim_K K[x]_d = \binom{n + d - 1}{n - 1}$$

   Dunque

   $$HP_{K[x]}(\lambda) = \sum_{d=0}^{\infty} \binom{n + d - 1}{n - 1} \lambda^d = \frac{1}{(1 - \lambda)^n}$$

2. Se $J = 1$, $K[x]/J = 0$ e dunque la dimensione dei polinomi di grado d nel quoziente è 0.

3. Procediamo per induzione sul grado del monomio. Se $|\alpha| = 1$, vale
$$K[x_1, \ldots, x_n]/x^\alpha \simeq K[y_1, \ldots, y_{n-1}]$$
e dunque
$$HP_{K[x]/x^\alpha}(\lambda) = \frac{1}{(1-\lambda)^{n-1}} = \frac{1-\lambda}{(1-\lambda)^n}$$
Supponiamo ora $|\alpha| \geq 1$ e $x^\alpha = x_1^{\alpha_1} \ldots x_n^{\alpha_n}$. Possiamo supporre che $\alpha_1 \geq 1$. Utilizzando il lemma 6.18 con $F = x_1$, otteniamo
$$HP_{K[x]/x^\alpha}(\lambda) = \lambda HP_{K[x]/(x^\alpha : x_1)}(\lambda) + HP_{K[x]/x_1}(\lambda)$$
Dato che $(x^\alpha : x_1) = x_1^{\alpha_1 - 1} x_2^{\alpha_2} \ldots x_n^{\alpha_n}$, per ipotesi induttiva vale
$$HP_{K[x]/(x^\alpha : x_1)}(\lambda) = \frac{1 - \lambda^{|\alpha|-1}}{(1-\lambda)^n}$$
Per quanto visto nel caso base, si ottiene
$$HP_{K[x]/x^\alpha}(\lambda) = \lambda \frac{1 - \lambda^{|\alpha|-1}}{(1-\lambda)^n} + \frac{1-\lambda}{(1-\lambda)^n} = \frac{1-\lambda^{|\alpha|}}{(1-\lambda)^n}$$

4. Consideriamo $J = (x^\alpha I)$. Utilizzando il lemma 6.18 con $F = x^\alpha$ otteniamo
$$HP_{K[x]/J}(\lambda) = \lambda^{|\alpha|} HP_{K[x]/(x^\alpha I : x^\alpha)}(\lambda) + HP_{K[x]/x^\alpha}(\lambda)$$
Dato che $(x^\alpha I : x^\alpha) = I$, otteniamo dal punto precedente
$$HP_{K[x]/J}(\lambda) = \frac{\lambda^{|\alpha|} < I > + 1 - \lambda^{|\alpha|}}{(1-\lambda)^n}$$

5. Utilizzando il lemma 6.18 con $F = x^\alpha$, otteniamo
$$HP_{K[x]/(I, x^\alpha)}(\lambda) = HP_{K[x]/I}(\lambda) - \lambda^{|\alpha|} HP_{K[x]/(I : x^\alpha)}(\lambda)$$
da cui
$$< I, x^\alpha > = < I > - \lambda^{|\alpha|} < I : x^\alpha >$$

□

La divisione per un monomio è un'operazione semplice se i generatori sono tutti monomi, poiché allora vale che
$$(x^{\alpha_1}, \ldots, x^{\alpha_s}) : x^\alpha = (x^{\alpha_1} : x^\alpha) + \cdots + (x^{\alpha_s} : x^\alpha)$$
e la divisione tra monomi è
$$(x^\beta : x^\alpha) = \left(x^\beta / gcd(x^\beta, x^\alpha) \right)$$
L'algoritmo per calcolare $< I >$, dunque, prende in input un insieme di monomi L che generano I, e ad ogni passo agisce con un monomio effettivo, ossia che renda i generatori più semplici.

Interriduciamo L
if $L = (x^\alpha, x^\beta)$ **then return** $1 - \lambda^{|\alpha|} + \lambda^{|\beta|}(1 - \lambda^{|\alpha-\beta|})$
end if
Scegliamo un x^α effettivo per L
return $< (L, x^\alpha) > + \lambda^{|\alpha|} < L : x^\alpha >$

Per scegliere l'elemento effettivo, si considera una delle variabili presente nella maggior parte dei termini e si prende uno dei due seguenti:

- Consideriamo tre termini in cui tale variabile compare. L'elemento effettivo sarà il GCD dei tre termini.
- Consideriamo due termini in cui tale variabile compare. L'elemento effettivo sarà la più piccola potenza della variabile scelta che compare nei due termini.

Non c'è una scelta più conveniente dell'altra, poiché esistono esempi in cui funziona male una, mentre l'altra bene.

Dato un modulo M dentro $K[x]^r$, con un ordinamento pos+to, possiamo definire una funzione (e quindi una serie) di Hilbert tramite

$$H : \mathbb{N} \to \mathbb{N} \qquad H(d) = \dim_K M_d$$

dove stiamo mettendo su $K[x]^r$ una gradazione compatibile con l'ordinamento.

Estraendo una base di Gröbner di M, possiamo spezzarlo in sottomoduli $M_i \subseteq K[x]$, rappresentati dalle colonne con lo stesso numero di zeri nelle prime componenti.

$$M \to \begin{pmatrix} I_1 & 0 & \dots & 0 \\ * & I_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ * & * & * & I_r \end{pmatrix}$$

Per additività della serie di Hilbert, avremo che

$$\sum_i HP_{k[x]/I_i} = HP_{k[x]^r/M} = HP_{k[x]^r} - HP_M = rHP_{k[x]} - HP_M$$

e dunque

$$HP_M = \frac{r - \sum < I_i >}{(1 - \lambda^n)}$$

6.6.1 Ideali Omogenei

Dati due ideali omogenei I, J nel solito anello di polinomi, allora $I \cap J$, $I : J$, $I :^\infty J$ sono ideali omogenei. Sappiamo inoltre che gli ideali monomiali sono omogenei, ma purtroppo le Syzigie fatte su dei monomi non sono un ideale monomiale in generale, e neanche omogeneo.

Se però prendiamo $(f_1, \ldots, f_s)$, con ogni f_i omogeneo, c'è un trucco per far tornare le Syzigie omogenee: basta mettere i giusti gradi alle componenti del modulo delle Syzigie.

$$\begin{array}{ccc} deg(f_1) & \to & \begin{pmatrix} f_1 & \ldots & f_s \\ 1 & \ldots & 0 \end{pmatrix} \\ \vdots & & \begin{pmatrix} \vdots & \ddots & \vdots \end{pmatrix} \\ deg(f_s) & \to & \begin{pmatrix} 0 & \ldots & 1 \end{pmatrix} \end{array}$$

Questo garantisce che i vettori siano omogenei dall'inizio alla fine.

Lemma 6.20. Dati $I \subseteq J$ ideali omogenei, allora

$$H_{K[x]/I}(d) \geq H_{K[x]/J}(d) \quad \forall\, d$$

e vale l'uguale se e solo se $I = J$

Dimostrazione. La disuguaglianza è ovvia. Se poniamo $HP_{K[x]/I} = HP_{K[x]/J}$, e $I \neq J$, allora esiste un minimo d naturale per cui $I_d \neq J_d$, dove $I_d = BGR_d(I)$ sono gli elementi della base di Gröbner ridotta di I con grado minore o uguale a d. allora $f \in I_d$ se e solo se è di grado minore o uguale a d e si riduce a zero tramite $BGR_d(I)$. Dato che $I_d \neq J_d$, allora esiste $f \in J_d$ che non appartiene a I_d, ossia si riduce a $f' \neq 0$. Dunque $Lt(f') \notin Lt(J_d)$, ma allora $H_{K[x]/I}(d) \neq H_{K[x]/J}(d)$, assurdo. □

Questo risultato ci permette di modificare leggermente il calcolo di una base di Gröbner di un ideale. Infatti, calcolare $H_{I/I_d}(d+1)$ ci dà un limite superiore al numero di polinomi di grado $d+1$ che non si annullano nella base di Gröbner, ossia fornisce un controllo per il calcolo degli S-polinomi. Per ottenere questo numero, basta calcolare

$$H_{I/I_d}(d+1) = H_{K[x]/I_d}(d+1) - H_{K[x]/I}(d+1)$$

che sarà positivo grazie al lemma sopra.

6.7 Risoluzione Libera Minimale

Dato un modulo $M \subseteq K[x]^r$, vorremmo trovare una sua risoluzione libera, ossia una successione esatta di moduli liberi su $K[x]$ fatta come

$$\cdots \to M'' \to M' \to M \to 0 \qquad M^{(i)} = K[x]^{r_i}$$

Un metodo per farlo è prendere le Syzigie di una base di Gröbner del modulo ad ogni passaggio, ossia dato $L = (f_1, \ldots, f_s)$ BGR di M, allora $Syz(L) \subseteq K[x]^s = M'$, e definiamo la mappa

$$\begin{array}{ccc} M' & \longrightarrow & M \\ e_i & \longmapsto & f_i \end{array}$$

Induttivamente, detta $L^{(i)} = (g_1, \ldots, g_k)$ la base di Gröbner del modulo delle sizigie in $M^{(i)}$, si pone $M^{(i+1)} = K[x]^k$ e si calcola la base di Gröbner $L^{(i+1)}$ di $Syz(L^{(i)})$.

Esempio. Consideriamo l'ideale $I = (x, y)$ in $K[x, y]$. La risoluzione libera minimale sarà

$$0 \to K[x,y] \xrightarrow{f} K[x,y]^2 \xrightarrow{g} I \to 0 \quad f(1) = -ye_1 + xe_2 \quad g(e_1) = x, g(e_2) = y$$

Un vantaggio di questo metodo è che se prendiamo $M = I \subseteq K[x]$ omogeneo, allora il numero di moduli nella risoluzione è finito, ed è minore del numero di variabili dell'anello.
Notiamo che le sizigie di un anello dipendono dai suoi generatori, soprattutto dal loro numero, poiché questo fa cambiare le dimensioni dei moduli della risoluzione libera minimale. Per fissare un metodo, si prende una base di Gröbner (Ridotta) dell'ideale, e si estraggono le Syzigie di quei generatori, quindi si calcola la base di Gröbner delle syzigie e si continua fino alla fine della risoluzione libera, ossia quando le Syzigie si annulleranno.

Dal punto di vista algoritmico, questo corrisponde a prendere una BGR dell'ideale di partenza, aggiungere una matrice identità, calcolare le sizigie ed iterare. Se J è l'ideale di partenza,

$$\big(RBG(J)\big) \to \begin{pmatrix} RBG(J) \\ I \end{pmatrix} \to \begin{pmatrix} RBG(J) & 0 \\ * & RBG(Syz) \end{pmatrix} \to \begin{pmatrix} RBG(Syz) \\ I \end{pmatrix} \to \cdots$$

Un'alternativa consiste nell'aggiungere delle variabili ausiliarie S, T di grado 0, per indicare a che modulo della risoluzione siamo, e in quale componente del modulo. Se inoltre l'ideale iniziale J è omogeneo, si può aggiungere una variabile D di grado 1 per mantenere l'omogeneità dei termini nei diversi passaggi.

L'algoritmo complessivo non è semplice da spiegare nei dettagli, dunque diamo l'idea generale.

- l'ordinamento avviene tramite la matrice

$$\begin{pmatrix} 1 & \dots & 1 & 0 & 1 & 0 \\ 0 & \dots & 0 & -1 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 & 1 \\ & & & & & \\ & Id & & & 0 & \\ & & & & & \end{pmatrix}$$

  dove le ultime tre colonne sono riferite rispettivamente a S, D, T, mentre le prime alle variabili x. Notiamo che, dato che gli elementi sono omogenei, saranno separati prima per S e poi per T.

- Si incomincia con gli elementi dell'ideale J, e si prova ad estrarre una base di Gröbner con un ordinamento DegLex. L'idea è che finchè riusciamo a ridurre attraverso l'algoritmo classico, non ci conviene fare le Syzigie. Notiamo che nel ridurre NON possiamo moltiplicare per S o per T, perché queste indicano la posizione dell'elemento, e non il suo valore o grado.

- Dato che comunque vogliamo fare al massimo un passo alla volta, vorremmo che un singolo termine, lungo tutto l'algoritmo, possa avere al massimo due potenze diverse, ma consecutive, di S. L'idea è che vogliamo calcolare un set di generatori delle Syzigie, prima di calcolarne di nuovo le syzigie. Dunque, definiamo lo "scarto" di un termine come il numero di potenze diverse di S che appaiono.

- Se un polinomio non è più riducibile, ossia tutti i suoi S-polinomi sono zero, ed è ridotto con gli altri elementi che sono presenti, allora la sua "testa", ossia i suoi termini con grado di S minore, apparterranno alla relativa base di Gröbner, dunque lo segnamo. Ci si presentano due possibilità:
  - Se lo scarto è uno, allora non facciamo niente
  - Se lo scarto è zero, possiamo cominciare ad estrarne le Syzigie, attacandoci una "coda": se il termine di testa è $cx^aS^iD^kT^l$, allora aggiungiamo $S^{i+1}D^{k+a}T^b$. Essenzialmente, stiamo aggiungendo una colonna della matrice identità, infatti la S ci dice che siamo passati al prossimo modulo, e aggiungiamo all'esponente della D il grado delle variabili x in modo da farlo rimanere omogeneo. L'esponente da dare a T indica il numero della colonna dell'identità aggiunta.
- Alla fine, gli elementi segnati comporranno le basi di groebner relative ai moduli indicati dalle S, dunque le BGR delle varie syzigie.

6.8 Equazioni in $\mathbb{Z}/n\mathbb{Z}$

Vorremmo ora trovare le radici di un polinomio con coefficienti in $\mathbb{Z}/n\mathbb{Z}$. Se n è piccolo, è meglio farlo a mano, dunque supponiamo $n >> 0$. In generale, però, n avrà una fattorizzazione, dunque si spezzerà in pezzi del tipo $\mathbb{Z}/p^\alpha Z$, dunque possiamo trovare le soluzioni lì e rimontare con il teorema cinese del resto. In realtà troviamo più soluzioni di quelle che in realtà esistono, ma a questo punto basta provarle.

Per prima cosa, consideriamo il caso $\mathbb{Z}/p\mathbb{Z}$. Notiamo che non conviene utilizzare l'algoritmo di fattorizzazione perché abbiamo bisogno solo dei termini lineari. Calcoliamo allora la parte lineare mediante il massimo comune divisore $g = (f, x^p - x)$. Se g ha ancora grado 0,1 o 2, è possibile ricavare a mano tutte le radici (a meno di saper calcolare l'inverso, e la radice di ogni classe di resto). Altrimenti, si prova a fattorizzare ancora tramite

$$\left(f, (x+a)^{\frac{p-1}{2}} - 1\right)$$

al variare di a tra le classi di resto. La probabilità di scegliere un a sbagliato, è $2^{1-deg(f)}$. Per trovare la molteplicità, è sufficiente provare a dividere per le radici trovate in questo modo.

Trattiamo ora il caso di $\mathbb{Z}/p^\alpha Z$. Dato che sappiamo risolvere il problema per $\alpha = 1$, procediamo per induzione e supponiamo di conoscere le radici $\alpha_1, \alpha_2, \ldots, \alpha_k$ in $\mathbb{Z}/p^{\alpha-1}Z$. Vogliamo scoprire quali di queste si sollevano a radici in $\mathbb{Z}/p^\alpha Z$.

Lemma 6.21. Sia z una radice di $f(x) \in \mathbb{Z}[x]$ su $\mathbb{Z}/p^{\alpha-1}Z$ e sia $f'(x)$ la derivata di $f(x)$. Allora

- $f'(z) \neq 0 \pmod p \implies$

$$\exists!\, y \in \mathbb{Z}/p^\alpha Z \;:\; f(y) \equiv 0 \pmod{p^\alpha} \quad y \equiv z \pmod{p^{\alpha-1}}$$

- $f'(z) = 0 \pmod p \quad f(z) = 0 \pmod{p^\alpha} \implies$

$$\forall s \in \mathbb{Z}/p\mathbb{Z} \quad f(z + sp^{\alpha-1}) \equiv 0 \pmod{p^\alpha}$$

- $f'(z) = 0 \pmod{p} \quad f(z) \neq 0 \pmod{p^\alpha} \implies$

$$\nexists\, y \in \mathbb{Z}/p^\alpha Z \;:\; f(y) \equiv 0 \pmod{p^\alpha} \quad y \equiv z \pmod{p^{\alpha-1}}$$

Dimostrazione. Cerchiamo $x = z + sp^{\alpha-1} \in \mathbb{Z}/p^\alpha Z$ che risolva $f(x) \equiv 0 \pmod{p^\alpha}$ e $s \in \mathbb{Z}/p\mathbb{Z}$. Ma

$$f(z + sp^{\alpha-1}) \equiv f(z) + sp^{\alpha-1} f'(z) \equiv 0 \pmod{p^\alpha} \iff$$

$$\iff {}^{f(z)}\!/_{p^{\alpha-1}} + sf'(z) \equiv 0 \pmod{p}$$

e da qui segue la tesi

□

Dato che le radici in p^α si riducono a radici in $p^{\alpha-1}$, abbiamo trovato tutte le radici.