Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE
Corso di Laurea in Matematica

IL TEST DI PRIMALITÀ AKS

Tesi di Laurea in Algebra

Relatore:
Chiar.mo Prof.
Mirella Manaresi

Presentata da:
Sara Giostra

I Sessione
2014/2015

Indice

Introduzione

Per il Teorema Fondamentale dell'Aritmetica, ogni intero si scrive in modo unico come prodotto di potenze di primi. Pertanto i numeri primi rappresentano i "mattoni" con cui si costruiscono tutti gli interi. Questo teorema era sostanzialmente già noto anche ai matematici Greci e ciò spiega perché, fin dall'antichità, siano stati cercati metodi per stabilire se un dato numero è primo. Il risultato più famoso in questo ambito è il Crivello di Eratostene, basato sull'osservazione che, se un numero n non è primo, allora è divisibile per un intero $\leq \sqrt{n}$. Questo risultato, importante dal punto di vista teorico, non è molto efficace nella pratica appena il numero n diventa abbastanza grande.
A partire dal risultato di Eratostene, numerosi sono stati i tentativi di trovare test di primalità efficienti. I test di primalità possono essere deterministici, quando stabiliscono con certezza se un numero è primo o meno, o probabilistici, nel qual caso non si ha la certezza che un numero che supera il test sia primo, ma la probabilità che non lo sia può essere resa davvero bassa.
Per molto tempo si è cercato di stabilire se esistesse un test di primalità deterministico e con complessità computazionale polinomiale, quindi eseguibile "in tempi ragionevoli". Un ruolo di grande importanza in questa ricerca fu giocato dal Piccolo Teorema di Fermat, che, nel 1975, ispirò G. L. Miller, permettendogli di trovare un test di primalità polinomiale e deterministico sotto l'ipotesi che sia vera una congettura sulla distribuzione dei numeri primi, chiamata Congettura di Riemann Generalizzata.
L'anno successivo, il matematico Rabin, basandosi sul test di Miller, ottenne un algoritmo polinomiale e probabilistico, che non utilizza l'ipotesi sulla validità della Congettura di Riemann Generalizzata.
Un ulteriore importante risultato fu raggiunto nel 1983 da L. M. Adleman, C. Pomeran-

ce e R. S. Rumley, i quali fornirono un test di primalità deterministico con complessità subesponenziale.

Un vero algoritmo deterministico e polinomiale è stato trovato solo nel 2002 da Manindra Agrawal, Neeraj Kayal e Nitin Saxena, tre giovani ricercatori dello Indian Institute of Technology di Kanpur. Questo test di primalità è oggi noto come test AKS e fu presentato per la prima volta nell'articolo "PRIMES is in P", pubblicato negli Annals of Mathematics. Tale risultato si basa su una generalizzazione del Piccolo Teorema di Fermat all'anello dei polinomi $\mathbb{Z}_p[x]$.

Lavori successivi di vari autori hanno migliorato il risultato di Agrawal, Kayal e Saxena, abbassandone la complessità computazionale.

Anche se nella crittografia a chiave pubblica continuano ad essere utilizzati test di primalità probabilistici, che derivano dal teorema di Miller-Rabin, l'algoritmo AKS ha fornito una svolta significativa, dal punto di vista concettuale, in un settore di grande interesse pratico. Infatti la sicurezza congetturale del sistema di crittografia a chiave pubblica RSA è basato sulla diversa complessità computazionale dell'operazione di criptazione rispetto a quella di decriptazione e il teorema AKS rafforza tale ipotesi congetturale.

Questo lavoro è diviso in tre parti: nel primo capitolo verranno richiamati concetti di algebra e verrà spiegato il significato di complessità computazionale di un algoritmo, nel secondo verrà esposto il teorema alla base dell'algoritmo AKS e nel terzo sarà presentato l'algoritmo e analizzata la sua complessità.

Capitolo 1

Notazioni e prerequisiti

1.1 Notazioni

Dato un gruppo $(G, *)$ indicheremo con $|g|$ l'ordine di $g \in G$.
Dato $(A, +, \cdot)$ anello e $a \in A$, indicheremo con $|a|$ l'ordine di a in $(A,+)$ e con 1_A l'unità di A.
Se a e b sono interi ed esiste $k \in \mathbb{Z}$ tale che $a = kb$, diremo che b divide a e scriveremo $b|a$. Scriveremo $b \nmid a$ quando, $\forall\ k \in$ Z, $a \neq kb$.
Sia $a \in \mathbb{Z}$, p un primo e $k \in \mathbb{N}$. Scriveremo $p^k || \ a$ se $p^k |\ a$ ma $p^{k+1} \nmid\ a$.
Dati due numeri interi a e b ed un intero positivo m, diremo che a è congruo a b modulo m se $m \mid a - b$ e scriveremo $a \equiv b \bmod m$.
Per n intero positivo, indicheremo con $\varphi(n)$ il numero degli interi non negativi minori o uguali ad n che sono coprimi con n. La funzione φ si chiama funzione di Eulero.
Dato n intero maggiore o uguale di 2, indicheremo con $\mathbb{Z}_n$ l'insieme delle classi di resto modulo n, il quale è un anello con le operazioni $[a]+[b] = [a+b]$, $[a][b] = [ab]$. L'elemento neutro rispetto alla somma è [0] e l'unità è [1]. Si ha che $\mathbb{Z}_n = \{[0], [1], ..., [n-1]\}$. $\mathbb{Z}_n^*$ rappresenterà il gruppo moltiplicativo formato dagli elementi invertibili di $(\mathbb{Z}_n, \cdot)$.
Dati n un intero positivo e a un intero primo con n, denoteremo con $\mathrm{G_{SS}}(n, a)$ il minimo intero positivo m tale che $a^m \equiv 1 \pmod{n}$, ovvero l'ordine di [a] quale elemento di $\mathbb{Z}_n^*$. $\mathrm{G_{SS}}(n, a)$ si chiama *gaussiano* di n rispetto ad a.
$\mathbb{Z}_n[x]$ sarà l'anello dei polinomi a coefficienti in $\mathbb{Z}_n$.

1.2 Richiami

Proposizione 1.2.1. *Una classe di resto* [a] *in* $\mathbb{Z}_n$ *è invertibile se e solo se*

$$\mathrm{MCD}(a, n) = 1.$$

Dimostrazione. $[a]$ è invertibile se e solo se esiste un intero b tale che $ab \equiv 1 \mod n$, ossia $ab - 1 = kn$ per un certo $k \in Z$.
Supponiamo che [a] sia invertibile. Allora $d = \mathrm{MCD}(a, n) | \ 1$, cioè $d = 1$. Viceversa, se $d = \mathrm{MCD}(a, n) = 1$, allora per l'identità di Bézout esistono u e v tali che $ua + vn = 1$. Prendendo $b = u$, si ha la tesi. □

Corollario 1.2.2. *Se* n *è un numero primo l'anello* $\mathbb{Z}_n$ *è un campo.*

Teorema 1.2.3 (Teorema di Eulero)**.** *Sia* $a \in \mathbb{Z}$ *e* $n \in \mathbb{Z}$*, con* $\mathrm{MCD}(a, n) = 1$*. Allora*

$$a^{\varphi(n)} \equiv 1 \qquad \mod n.$$

Dimostrazione. Poiché a è coprimo con n, per la proposizione (1.2.1) [a] è invertibile in $\mathbb{Z}_n$ e il gruppo moltiplicativo $\mathbb{Z}_n^*$ ha ordine $\varphi(n)$. Pertanto

$$[a]^{\varphi(n)} = [1],$$

da cui $[a^{\varphi(n)}] = [1]$, ovvero $a^{\varphi(n)} \equiv 1 \mod n$. □

Come caso particolare del Teorema di Eulero, si trova

Teorema 1.2.4 (Piccolo Teorema di Fermat)**.** *Siano* p *un numero primo e* a *un intero non divisibile per* p*. Allora*

$$a^{p-1} \equiv 1 \mod p. \tag{1.1}$$

Osservazione 1. Se nel teorema 1.2.4 a è un intero divisibile per p, allora

$$a^p \equiv a \mod p$$

Il Piccolo Teorema di Fermat è alla base di numerosi test di primalità. Citiamo di seguito quello probabilistico di Miller-Rabin, il più utilizzato nelle applicazioni.
Ricordiamo innanzitutto che i test probabilistici T consistono in una successione, eventualmente finita, di test $\{T_m\}_{m\in\mathbb{N}}$, per i quali esiste una successione $\{\epsilon_m\}_{m\in\mathbb{N}}$ di numeri reali positivi minori di 1 convergente a 0 , tale che se un numero intero positivo n non passa un test T_m allora non è primo, mentre la probabilità che un numero intero positivo n passi i test $T_1, ..., T_m$ e non sia primo è minore di ϵ_m.
Il test di Miller-Rabin è basato sul concetto di *pseudoprimo forte*. Vediamo allora la seguente

Definizione 1.1. Sia n un numero intero positivo dispari non primo e $b < n$ un numero intero positivo tale che $MCD(b, n) = 1$. Scriviamo $n = 2^s t + 1$, con t dispari. Il numero n si dice *pseudoprimo forte in base* b se

$$b^t \equiv 1 \pmod{n},$$

oppure se esiste un numero intero non negativo $r < s$ tale che

$$b^{2^r t} \equiv -1 \pmod{n}.$$

Osservazione 2. Dato n un numero intero positivo dispari non primo, i numeri positivi $b < n$ tali che $MCD(b, n) = 1$ e tali che n sia uno pseudoprimo forte in base b sono non più di un quarto di tutti i numeri positivi $b < n$ tali che $MCD(b, n) = 1$.
Per la dimostrazione rimandiamo a [2], prop. 6.1.20 pag.276.

Presentiamo quindi il test di Miller-Rabin. Fissiamo un intero dispari $n > 1$. Scriviamo $n = 2^s t + 1$ con t dispari. Il test T_1 consiste nei seguenti passi:

1. scegliamo a caso un intero b_1 con $1 < b_1 < n$ e calcoliamo il $MCD(b_1,n)$;
2. se $MCD(b_1, n) > 1$ allora n non è primo e abbiamo concluso;
3. se $MCD(b_1, n) = 1$ calcoliamo b_1^t modulo n. Se $b_1^t \equiv \pm 1 \pmod{n}$, n è primo o è uno pseudoprimo forte in base b_1;
4. se $b_1^t \not\equiv -1 \pmod{n}$ calcoliamo b_1^{2t} modulo n. Se $b_1^{2t} \equiv -1$ modulo n, allora n è primo oppure è uno pseudoprimo forte in base b_1;

5. se $b_1^{2t} \not\equiv -1 \pmod{n}$ procediamo ancora nel medesimo modo. Se tutte le potenze successive $b_1^{2^r t}$, per $r = 1, ..., s-1$, non sono mai congrue a -1 modulo n, n non è primo (cfr [2], Osservazione 6.1.15 pag. 265). Altrimenti n è uno pseudoprimo forte in base b_1.

Il test T_m si definisce ricorsivamente nello stesso modo:

1. scegliamo a caso un intero b_m con $1 < b_m < n$ con b_m diverso da $b_1, ..., b_{m-1}$ e calcoliamo il $MCD(b_m,n)$;

2. se $MCD(b_m, n) > 1$ allora n non è primo e abbiamo concluso;

3. se $MCD(b_m, n) = 1$ si calcola b_m^t modulo n e si procede come dal passo (3) di T_1 in poi.

Per l'osservazione 2, la probabilità che n non sia primo ma passi i test $T_1, ..., T_m$ è minore di $1/4^m$.

Diamo ora una caratterizzazione dei numeri primi.

Proposizione 1.2.5. *Un intero positivo n, $n \geq 2$ è numero primo se e solo se $\forall$ k intero positivo tale che $1 < k < p$, si ha che n divide $\binom{n}{k}$.*

Dimostrazione. Supponiamo n primo. Per definizione di coefficiente binomiale, vale che

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{n(n-1)\cdot\cdot\cdot(n-k+2)(n-k+1)}{k!},$$

da cui, considerando anche che per ipotesi $1 < k < n$, si ha

$$n(n-1)\cdot\cdot\cdot(n-k+2)(n-k+1) = \binom{n}{k}\cdot k!.$$

Allora n divide $\binom{n}{k}\cdot k!$ ed essendo primo deve dividere uno dei due fattori. Esso non può però dividere $k!$ poiché ogni fattore di quest'ultimo è, per ipotesi, minore di p, da cui $n \mid \binom{n}{k}$.
Il viceversa viene provato per assurdo. Se n non è primo, n si può scrivere nella forma

$n = p^k m$ per un qualche primo p, $k \in \mathbb{N}$, $m \geq 2$ e $p^k || n$, ovvero $p \nmid m$.
Allora vale

$$\binom{n}{p} = \frac{n!}{p!(n-p)!} = \frac{n(n-1)\cdot\cdot\cdot(n-p+2)(n-p+1)}{p!} =$$

$$= \frac{p^k m(n-1)\cdot\cdot\cdot(n-p+1)}{p!} = \frac{p^{k-1}m(n-1)\cdot\cdot\cdot(n-p+1)}{(p-1)!}$$

da cui si vede che $p^{k-1}||\binom{n}{p}$, quindi $n = p^k m$ non divide $\binom{n}{p}$, il che costituisce un assurdo.

□

Proposizione 1.2.6. *Sia $n \in \mathbb{N}$. Allora $\forall\ k\ \in \mathbb{N}$, $k \leq 2n$, vale $\binom{2n}{k} \leq \binom{2n}{n}$.*

Dimostrazione. Supponiamo inizialmente che $0\ \leq k\ \leq n$.
Si ha $n \leq 2n-k$, $n-1 \leq 2n-k-1$,..., $k+1 \leq n+1$. Moltiplicando membro a menbro, si ottiene

$$n\cdot(n-1)\cdot\cdot\cdot(k+2)\cdot(k+1) \leq (2n-k)\cdot(2n-k-1)\cdot\cdot\cdot(n+1).$$

Moltiplicando entrambi i membri per $n!\cdot k!$, segue che $n!\cdot n! \leq (2n-k)!\cdot k!$. Pertanto

$$\binom{2n}{n} = \frac{(2n)!}{n!\cdot n!} \geq \frac{(2n)!}{k!\cdot(2n-k)!} = \binom{2n}{k}.$$

Ma $\binom{2n}{k} = \binom{2n}{2n-k}$, quindi la disuguaglianza è dimostrata anche per $n \leq k \leq 2n$.

□

Lemma 1.2.7. *Sia B un numero intero positivo e* $\mathrm{mcm}(B)$ *il minimo comune multiplo degli interi positivi minori o uguali a B. Per $B \geq 7$ si ha che* $\mathrm{mcm}(B) \geq 2^B$.

Dimostrazione. Una verifica diretta mostra che $\mathrm{mcm}(7) = 420 \geq 128 = 2^7$ e $\mathrm{mcm}(8) = 840 \geq 256 = 2^8$. Pertanto, assumiamo $B \geq 9$.
Siano m e n interi positivi, con $1 \leq m \leq n$.
Poniamo

$$I_{m,n} = \int_0^1 x^{m-1}(1-x)^{n-m}dx$$

e calcoliamo questo integrale in due modi diversi.
La prima maniera consiste nell'utilizzare ripetutamente il metodo di integrazione per parti, ottenendo

$$I_{m,n} = \int_0^1 x^{m-1}(1-x)^{n-m}dx = \left[\frac{x^m}{m}(1-x)^{n-m}\right]_0^1 + \int_0^1 \frac{x^m}{m}(n-m)(1-x)^{n-m-1}dx =$$

$$= \frac{n-m}{m}\int_0^1 x^m(1-x)^{n-m-1}dx = \frac{(n-m)(n-m-1)}{(m+1)m}\int_0^1 x^{m+1}(1-x)^{n-m-2}dx =$$

$$= \frac{(n-m)(n-m-1)\cdot\cdot\cdot 2\cdot 1}{(n-1)(n-2)\cdot\cdot\cdot m}\int_0^1 x^{n-1}dx = \frac{(n-m)(n-m-1)\cdot\cdot\cdot 2\cdot 1}{n\cdot(n-1)(n-2)\cdot\cdot\cdot m} = \frac{1}{m\cdot\binom{n}{m}}.$$

Calcoliamo ora l'integrale sviluppando il binomio. Si ha

$$I_{m,n} = \int_0^1 x^{m-1}(1-x)^{n-m}dx = \int_0^1 \sum_{r=0}^{n-m}\binom{n-m}{r}(-1)^r x^r dx =$$

$$= \sum_{r=0}^{n-m}\binom{n-m}{r}(-1)^r\int_0^1 x^{m+r-1}dx = \sum_{r=0}^{n-m}\binom{n-m}{r}(-1)^r\frac{1}{m+r}.$$

Poiché $m+r$ è intero, positivo e minore o uguale ad n, $\mathrm{mcm}(n)\cdot I_{m,n}$ è intero positivo, quindi per ogni intero positivo m minore o uguale ad n, $m\cdot\binom{n}{m}$ divide $\mathrm{mcm}(n)$.
In particolare, $n\cdot\binom{2n}{n}$ divide $\mathrm{mcm}(2n)$, da cui, a maggior ragione, $n\cdot\binom{2n}{n}$ divide $\mathrm{mcm}(2n+1)$. Allo stesso modo, $(n+1)\cdot\binom{2n+1}{n+1}$ divide $\mathrm{mcm}(2n+1)$.
Notiamo che

$$(2n+1)\cdot\binom{2n}{n} = \frac{(2n+1)\cdot(2n)!}{n!\cdot n!} = (n+1)\frac{(2n+1)!}{n!\cdot(n+1)!} = (n+1)\cdot\binom{2n+1}{n+1}.$$

Allora, dato che $2n+1$ e n sono primi tra loro, anche $n\cdot(2n+1)\cdot\binom{2n}{n}$ divide $\mathrm{mcm}(2n+1)$ e in particolare

$$\mathrm{mcm}(2n+1) \geq n\cdot(2n+1)\cdot\binom{2n}{n} \tag{1.2}$$

Sfruttando la proposizione 1.2.6, otteniamo

$$2^{2n} = (1+1)^{2n} = \sum_{k=0}^{2n}\binom{2n}{k} \leq (2n+1)\binom{2n}{n},$$

quindi, per la (1.2) ,

$$\mathrm{mcm}(2n+1) \geq n \cdot (2n+1) \cdot \binom{2n}{n} \geq n \cdot 2^{2n}.$$

Per $n \geq 2$ abbiamo subito che

$$\mathrm{mcm}(2n+1) \geq n \cdot 2^{2n} \geq 2^{2n+1}$$

e per $n \geq 4$

$$\mathrm{mcm}(2n+2) \geq \mathrm{mcm}(2n+1) \geq n \cdot 2^{2n} \geq 2^{2n+2},$$

cioè, sinteticamente, se $B \geq 9$

$$\mathrm{mcm}(B) \geq 2^B.$$

□

Richiamiamo ora alcuni concetti di teoria degli anelli e dei campi. Per una trattazione più approfondita rimandiamo a [2].

Definizione 1.2. In un anello commutativo unitario A, un ideale I è un sottogruppo di $(A,+)$ tale che, $\forall\ x \in I$ e $\forall\ a \in A$, si ha $ax \in I$.

Definizione 1.3. Dato $x \in A$, $(x) = \{ax/a \in A\}$ è un ideale, detto *ideale principale* generato da x.

Osservazione 3. Sia A un anello commutativo e ne sia I un ideale. In A si può definire la relazione di congruenza modulo I nel modo seguente: dati a e $b \in A$ si ha $a \equiv b$ modulo I se e solo se $a - b \in I$. Si verifica che la congruenza modulo I è una relazione di equivalenza, il cui insieme quoziente si denota con A/I.
L'insieme quoziente A/I può essere munito di una struttura di anello unitario definendo $\forall\ a, b \in A$ le operazioni

- $[a] + [b] = [a + b]$
- $[a] \cdot [b] = [a \cdot b]$

L'elemento neutro rispetto all'addizione è $[0]$ e l'unità di A/I è $[1]$.

Osservazione 4. Sia A un anello unitario. Si può considerare l'applicazione

$$f : \mathbb{Z} \longrightarrow A$$

così definita:

$$f(n) = \begin{cases} \underbrace{1 + ... + 1}_{n} & \text{per } n \text{ positivo} \\ 0 & \text{per } n = 0 \\ \underbrace{(-1) + ... + (-1)}_{n} & \text{per } n \text{ negativo} \end{cases}$$

L'applicazione f risulta un omomorfismo di anelli, ossia

$$f(n + m) = f(n) + f(m), \quad f(nm) = f(n)f(m).$$

Pertanto $f(\mathbb{Z})$ è un sottoanello di A, detto sottoanello fondamentale di A. Se f è iniettiva si dice che A ha caratteristica zero e il sottoanello fondamentale di A è isomorfo a $\mathbb{Z}$.
Se f non è iniettiva, allora esiste un minimo intero $n > 1$ tale che $f(n) = 0$. Esso si dice caratteristica di A e si indica con $\mathrm{car}A$.
Se A è un dominio di integrità, ovvero se non ha 0-divisori oltre lo 0, allora n è un numero primo. Infatti, se per assurdo $n = pq$ con p e q interi positivi minori di n, si avrebbe $0 = f(n) = f(pq) = f(p)f(q)$ e dunque avremmo o $f(p) = 0$ o $f(q) = 0$, il che è assurdo.

Definizione 1.4. Supponiamo di avere due campi K e F, con K sottocampo di F. Si dice che F è un'estensione di K. Se $b_1, ..., b_n$ sono elementi di F, si indica con $K(b_1, ..., b_n)$ il minimo sottocampo di F contenente K e $b_1, ..., b_n$.

Definizione 1.5. Sia K un campo e sia $f(x)$ un polinomio di grado n su K. Un'estensione $K \subseteq F$ si dice campo di riducibilità completa o campo di spezzamento di $f(x)$ se esistono elementi $b_1, ..., b_n \in F$ e $a \in K$ tali che

- $f(x) = a(x - b_1) \cdot \cdot \cdot (x - b_n)$;
- $F = K(b_1, ..., b_n)$.

Osservazione 5. Si dimostra che dato un polinomio f è sempre possibile costruire un campo di spezzamento per f, che risulta essere la più piccola estensione di K che contiene tutte le radici di f. Si dimostra anche che il campo di spezzamento è unico a meno di isomorfismi (cfr [4]).

Definizione 1.6. Consideriamo il polinomio $f_n(x) = x^n - 1$ sul campo $\mathbb{K} = \mathbb{Z}_p$ con p numero primo, e sia F_n il campo di riducibilità completa di f. Le sue radici si dicono radici n-esime dell'unità.
É immediato verificare che l'insieme R_n delle radici n-esime dell'unità è un sottogruppo del gruppo moltiplicativo di F_n e che, se $m \mid n$, allora $R_m \subseteq R_n$.

Studiamo ora la struttura del gruppo R_n. Si ha:

Proposizione 1.2.8. *Se $\mathbb{K}$ ha caratteristica 0 oppure p con p primo con n, allora R_n è un gruppo ciclico di ordine n.*

Dimostrazione. Posto $f_n(x) = x^n - 1$, si ha $D(f_n(x)) = nx^{n-1}$. Nelle nostre ipotesi $n \neq 0$ in $\mathbb{K}$, pertanto $D(f_n(x))$ ha come una radice 0, con molteplicità $n-1$. Per il teorema di Ruffini $f_n(x)$ ha esattamente n radici distinte, ossia R_n ha ordine n.
Proviamo che R_n è ciclico. Per prima cosa, esaminiamo il caso in cui n sia potenza di un primo q, ossia $n = q^m$. Se in R_n non vi fosse alcun elemento di ordine n allora $\forall\, \xi \in R_n$ esisterebbe un $m_\xi < m$ tale che ξ ha ordine q^{m_ξ}. Sia $m' = \max\{m_\xi \mid \xi \in R_n\}$. Ovviamente $m' < m$ e quindi ciascun elemento di R_n avrebbe ordine $n' = q^{m'} < n$. Di conseguenza avremmo $R_n \subset R_{n'}$, il che è assurdo perché R_n ha ordine n, mentre $R_{n'}$ ha ordine $n' < n$.
Supponiamo ora che $n = q_1^{m_1} \cdot \cdot \cdot q_h^{m_h}$ e indichiamo con n_i il fattore $q_i^{m_i}$ per $i = 1, ..., h$. Si ha $R_{n_i} \subset R_n$ e R_{n_i} è ciclico di ordine n_i, poniamo che sia ξ_i un generatore di R_{n_i} per ogni $i = 1, ..., h$. Posto $\xi = \xi_1 \cdot \cdot \cdot \xi_n$, si verifica che l'ordine di ξ è proprio n. Supponiamo infatti che ξ abbia ordine $d < n$. Allora $d = q_1^{\mu_1} \cdot \cdot \cdot q_h^{\mu_h}$ con $\mu_1, ..., \mu_h$ interi non negativi e $\mu_i \leq m_i$, $i = 1, ..., h$, valendo per almeno un i la disuguaglianza stretta. Poniamo $d_i = q_i^{\mu_i}$ e $l_i = d/d_i$, $i = 1, ..., h$. Si ha $\xi^d = \prod_{i=1}^{h} (\xi_i^{d_i})^{l_i}$. Almeno uno degli elementi $\xi_i^{d_i}$ non è 1, supponiamo ciò accada per tutti e soli gli indici $i = 1, ..., k$, sicché $\xi^d = \prod_{i=1}^{k} (\xi_i^{d_i})^{l_i}$. Notiamo che $\xi_i^{d_i}$ ha ordine $\delta_i = n_i/d_i = q_i^{m_i - \mu_i}$ e $\delta_i \nmid l_i$ per $i = 1, ..., k$. Quindi per ogni $i = 1, ..., k$ si ha $\xi_i^d = (\xi_i^{d_i})^{l_i} \neq 1$. Infine si osservi che $\xi_1^d \xi_2^d \neq 1$ perché $R_{n_1} \cap R_{n_2} = (1)$. Similmente $\xi_1^d \xi_2^d \xi_3^d \neq 1$ perché $R_{n_1 n_2} \cap R_{n_3} = (1)$, e così via. □

Definizione 1.7. Sia $K \subseteq F$ un'estensione di campi e sia $u \in F$. Diciamo che u è algebrico su K se esiste un polinomio $f \in K[x]$ tale che $f \neq 0$ e $f(u) = 0$.

Definizione 1.8. Sia $K \subseteq F$ un'estensione di campi e sia u algebrico su K. Il polinomio minimo di u su K è l'unico polinomio di $K[x]$ monico, irriducibile, ovvero privo di divisori propri, e che si annulla su u.

Osservazione 6 (Campi finiti)**.** Se $\mathbb{K}$ è un campo finito, allora car $\mathbb{K} = p > 0$, dove p è un numero primo e $\mathbb{K}$ contiene $\mathbb{Z}_p$ come sottocampo fondamentale. $\mathbb{K}$ è un $\mathbb{Z}_p$-spazio vettoriale di dimensione finita. Se $n = \dim \mathbb{Z}_p$, allora K ha p^n elementi.

1.3 La complessità computazionale

É naturale domandarsi quanto costi un algoritmo, in termini di tempo necessario per seguirne i passi e portare a termine le operazioni che esso prevede. Rispondere a tale quesito significa studiare la complessità computazionale dell'algoritmo, la quale consente di distinguere tra:

- algoritmi eseguibili in "tempi ragionevoli";
- algoritmi non eseguibili in "tempi ragionevoli";
- algoritmi possibili solo in teoria, poiché richiedenti, per la loro esecuzione, un tempo in assoluto troppo lungo.

L'unità usata per calcolare la complessità di un algoritmo è l'operazione bit, la quale, in sostanza, corrisponde ad un'operazione elementare costituente l'algoritmo. Per operazione bit intendiamo una delle seguenti operazioni elementari:

1. addizione fra due cifre binarie;
2. sottrazione fra due cifre binarie;
3. moltiplicazione fra due cifre binarie;
4. divisione di un intero a due cifre binarie per una cifra binaria;

Ognuna delle quattro operazioni su interi qualsiasi è data da una successione di operazioni bit. Dunque ogni algoritmo è dato da una successione di operazioni bit. Pertanto:

Definizione 1.9. La complessità computazionale di un algoritmo che opera sugli interi è data dal numero di operazioni bit occorrenti per eseguirlo.

Notiamo che la complessità computazionale di un algoritmo non è un numero, ma una funzione. Infatti essa dipende, in generale, dagli interi su cui opera l'algoritmo. C'è da aspettarsi, quindi, che la complessità computazionale aumenti all'aumentare della grandezza degli interi su cui l'algoritmo opera.
Di norma non si è interessati al calcolo esatto della complessità di un algoritmo, bensì ad una sua ragionevole stima superiore. Per dare tale stima in modo efficiente si usa la notazione O, la cui importanza risiede nel fatto che essa evidenzia in modo semplice e diretto il comportamento asintotico di una data funzione.

Definizione 1.10. Siano f e g due funzioni a valori positivi, definite su un sottinsieme S di $\mathbb{R}$ contenente $\mathbb{N}$. Diremo che f è *dominata* da g se esistono costanti k e c in $\mathbb{R}$ tali che

$$f(x) \leq k \cdot g(x), \quad \textit{per ogni } x \in S, \textit{ con } x > c.$$

Indicheremo con *O(g)* l'insieme delle funzioni dominate da g, cioè quelle che crescono non più velocemente di g. Notiamo che, se $f \in O(g)$ e $g \in O(h)$, allora $f \in O(h)$. A partire dalle quattro operazioni elementari, si possono dare algoritmi per eseguire le quattro operazioni su due interi arbitrari a e b di al più m bit ciascuno, con m intero positivo. La complessità computazionale risulterà $O(m)$ per somma e sottrazione e $O(m^2)$ per moltiplicazione e divisione. Da ciò si possono dedurre algoritmi che operano su un polinomio f di grado d con coefficienti ad m bit, dove m e d sono interi positivi fissati. In questo caso si dimostra facilmente che $f = O(n^d)$.
Diremo che un algoritmo ha complessità computazionale $O(f(n))$ se, detto $T(n)$ il numero di operazioni binarie per eseguire l'input n, si ha $T(n) = O(f(n))$.
Notiamo che se esiste finito $\lim_{n\to\infty} \frac{f(n)}{g(n)}$, allora $f = O(g)$, cioè f e g sono dello stesso ordine.
Ricordando che, a meno di una costante, il numero di cifre binarie di n è all'incirca $\log_2 n$, diamo le seguenti definizioni:

Definizione 1.11. Un algoritmo per eseguire un calcolo su numeri interi si dice *di tempo polinomiale* o *polinomiale* se esiste un intero positivo d, detto ordine dell'algoritmo, tale

che il numero di operazioni bit necessarie per eseguire l'algoritmo su interi di lunghezza binaria al più k è $O(k^d)$. Equivalentemente, un algoritmo è polinomiale se $T(n) = O((log\ n)^d)$ (ovvero se il tempo di esecuzione cresce, all'incirca, come la d-esima potenza del numero di cifre binarie di n).

Definizione 1.12. Un algoritmo si dice *di tempo esponenziale* o *esponenziale* se il numero di operazioni bit necessarie per eseguire l'algoritmo su interi di lunghezza binaria al più k è dello stesso ordine di e^{ck}, per una costante $c > 0$.

Vediamo ora la differenza tra un algoritmo esponenziale e uno polinomiale. Supponendo di dover operare con numeri molto grandi, se un algoritmo è polinomiale possiamo aspettarci che esso sia effettuabile, magari utilizzando opportuni strumenti di calcolo, in un tempo ragionevole. Al contrario, se l'algoritmo è di tempo esponenziale, non si riesce, anche usando computer potenti, ad avere la risposta in tempi ragionevoli.

Esaminiamo ora alcuni esempi.
Un tipico esempio di algoritmo esponenziale è il Crivello di Eratostene, che serve a determinare se un numero intero è primo. Esso si fonda sul seguente risultato, basato sul Teorema Fondamentale dell'Algebra:

Proposizione 1.3.1. *Se un numero intero positivo n non è divisibile per nessun primo che è minore o uguale a $\sqrt{n}$, allora n è primo.*

Per determinare se un dato n è primo, il crivello procede in questa maniera. Si scrivono tutti i numeri minori o uguali ad n, a partire da 2, che viene sottolineato perché è primo. Poi si cancellano tutti i multipli di 2, perché non sono primi. Si sottolinea poi il primo numero non cancellato ossia 3, e si cancellano tutti i suoi multipli, sempre perché non sono primi, e così via, finché non ci siano numeri non cancellati minori o uguali a $\sqrt{n}$. Ebbene, tutti i numeri sottolineati, assieme a tutti quelli che non sono stati cancellati, forniscono la lista completa di tutti i numeri primi minori o uguali a n.
Questo metodo risulta inefficiente per numeri grandi, infatti ha complessità esponenziale.

Consideriamo adesso l'Algoritmo di Euclide, utile per trovare il massimo comun divisore di due interi positivi a e b, senza dover fattorizzare a e b nel prodotto di primi. Esso consiste nel dividere prima a per b, ottenendo un quoziente q_1 ed un resto r_1: $a = bq_1 + r_1$. Si ripete poi il procedimento, con b ed r_1 al posto di a e b: $b = r_1q_2 + r_2$. Si procede ricorsivamente, dividendo r_i per r_{i+1}, per $i \geq 1$, ottenendo $r_i = r_{i+1}q_{i+2} + r_{i+2}$, finché otteniamo un resto nullo. L'ultimo resto non nullo r_l è MCD(a,b).
Vale la seguente

Proposizione 1.3.2. *L'algoritmo di Euclide fornisce il massimo comun divisore di due interi positivi $a > b$ in un numero finito di passi, in un tempo $O(\log^3 a)$.*

Dimostrazione. Sappiamo già che l'algoritmo termina (poiché $r_{i+1} < r_i$) e che l'ultimo resto non nullo è MCD(a,b). Proviamo ora che i resti decrescono rapidamente, e precisamente $r_{i+2} < \frac{1}{2}r_i$. Infatti, poiché i resti sono decrescenti, tutti i quozienti sono strettamente positivi, pertanto

$$r_i = r_{i+1}q_{i+2} + r_{i+2} \geq r_{i+1} + r_{i+2} > 2r_{i+2}.$$

Ne segue che ci sono al più $2\lfloor \log_2 a \rfloor$ divisioni, e $2\lfloor \log_2 a \rfloor = O(\log\, a)$. Facendo il conto dettagliatamente, si ottiene

$$a > 2r_1 > 4r_3 > \cdots > 2^i r_{2i-1} > \cdots > 2^{\left\lfloor \frac{\hat{l}+1}{2} \right\rfloor} r_{\hat{l}} \geq 1,$$

dove r_l è l'ultimo resto non nullo e $\hat{l} = 2\left\lfloor \frac{l+1}{2} \right\rfloor - 1$, ovvero $\hat{l} = l$ se l è dispari e $\hat{l} = l - 1$ se l è pari. In ogni caso $\frac{l}{2} \leq \left\lfloor \frac{\hat{l}+1}{2} \right\rfloor \leq \lfloor \log_2 a \rfloor$.
Poiché ogni divisione coinvolge interi minori o uguali ad a, si esegue in $O(\log^2 a)$ operazioni bit. Di conseguenza il tempo totale necessario è $O(\log\, a)O(\log^2 a) = O(\log^3 a)$. □

Capitolo 2

Il Teorema di Agrawal, Kayal e Saxena

L'algoritmo AKS, che illustreremo nel prossimo capitolo, si basa sulla seguente versione polinomiale del Piccolo Teorema di Fermat.

Proposizione 2.0.3. *Siano $a \in \mathbb{Z}$, $n \in \mathbb{N}$, $n \geq 2$ tali che $MCD(a, n) = 1$. Allora n è un numero primo se e solo se*

$$(x + a)^n \equiv x^n + a \pmod{n}. \tag{2.1}$$

Dimostrazione. Scriviamo l'espansione del primo membro:

$$(x + a)^n = \sum_{k=0}^{n} \binom{n}{k} x^k a^{n-k}$$

Se n non è un numero primo, per la proposizione 1.2.5 uno dei coefficienti $\binom{n}{k}$, con $1 < k < n$, è non nullo modulo n. Pertanto $\binom{n}{k} a^{n-k} \not\equiv 0 \pmod{n}$, da cui la tesi.
Al contrario, se n è un numero primo, per la medesima proposizione si ha che $\binom{n}{k} \equiv 0 \pmod{n}$ per $1 < k < n$, per cui

$$(x + a)^n \equiv x^n + a^n \equiv x^n + a \pmod{n},$$

poiché, per il Piccolo Teorema di Fermat, $a^n \equiv a \pmod{n}$. □

L'identità (2.1) costituisce una caratterizzazione dei numeri primi e rappresenta, dunque, un test di primalità. Infatti, si potrebbe verificare se un numero n è primo scegliendo un $a \in Z$ tale che $MCD(a, n) = 1$ e vedere se vale la (2.1). Tuttavia questo test ha complessità esponenziale, dato che i coefficienti del polinomio $(x + a)^n$ sono $n + 1$, e il calcolo di ciascuno di essi può essere alquanto oneroso.
Pertanto, al fine di rendere l'algoritmo più efficiente, bisogna cercare di ridurre il numero dei coefficienti da calcolare, valutanto l'equazione (2.1) modulo un polinomio di grado relativamente basso. Cerchiamo un polinomio $f(x)$ di grado r, con r intero positivo non troppo grande, tale che, se

$$(x + a)^n = x^n + a \quad in\ \mathbb{Z}_n[x]/f(x), \tag{2.2}$$

allora vale la (2.1). Osserviamo che, se n è primo, vale la (2.1) e dunque anche la (2.2). In generale, però, non è detto che la (2.2) sia condizione sufficiente perché valga la (2.1) e quindi perché n sia primo. Vediamone, infatti, un esempio:

Esempio 2.1. Sia $f(x) = x^2 - 1$, allora si ha

$$(x + 3)^4 \equiv x^4 + 3 \quad in \quad \mathbb{Z}_4[x]/(x^2 - 1). \tag{2.3}$$

Infatti, in $\mathbb{Z}_4[x]$ vale

$$(x + 3)^4 = (x - 1)^4 = x^4 - 4x^3 + 6x^2 + 4x + 1 = x^4 + 2x^2 + 1$$

e, passando in $\mathbb{Z}_4[x]/(x^2 - 1)$, vale la (2.3). Dunque, pur essendo 4 non primo, la (2.2) è verificata.

Occorre quindi generalizzare la (2.1) opportunamente.
Il polinomio di grado r che utilizzeremo sarà $x^r - 1$ e quindi dimostreremo che, per r opportuno,

$$(x + a)^n = x^n + a \quad in\ \mathbb{Z}_n[x]/(x^r - 1) \tag{2.4}$$

implica la (2.1), da cui n primo per la proposizione 2.0.3.
Per ottenere un algoritmo polinomiale, dimostreremo che sia r sia il numero degli a per

cui la (2.1) deve essere verificata affinché si possa dire che n è primo, sono limitati da un polinomio in $\log n$.

Ricordiamo che per un elemento invertibile $[r]$ di $\mathbb{Z}_n$ abbiamo indicato con $\mathrm{G_{SS}}(r,n)$ l'ordine di $[r]$ in $\mathbb{Z}_n^*$.

Lemma 2.0.4. *Sia n un numero intero positivo. Allora esiste un intero $r \leq [16\log_2^5 n] + 1$ tale che $G_{SS}(r,n) > 4\log_2^2 n$.*

Dimostrazione. Siano $r_1, ..., r_t$ tutti i numeri positivi tali che $\mathrm{G_{SS}}(r_i, n) \leq 4\log_2^2 n$ per ogni $i = 1, ..., t$. Questo vuol dire che, per ogni $i = 1, ..., t$, vale

$$n^k \equiv 1 \pmod{r_i},$$

per qualche $k = 1, ..., [4\log_2^2 n]$, ossia ciascun r_i divide il prodotto

$$\prod_{i=1}^{[4\log_2^2 n]} (n^i - 1) < n^{16\log_2^4 n} \leq 2^{16\log_2^5 n}. \tag{2.5}$$

Poniamo $M = [16\log_2^5 n] + 1$. Tenendo presente la (2.5) e il lemma 1.2.7 si ha

$$\mathrm{mcm}(r_1, ..., r_t) < 2^M \leq \mathrm{mcm}(1, 2, ..., M).$$

Infatti, se così non fosse, si avrebbe che $\{1, ..., M\} \subset \{r_1, ..., r_t\}$ e, di conseguenza, varrebbe

$$\mathrm{mcm}(1, 2, ..., M) \leq \mathrm{mcm}(r_1, ..., r_t) < \mathrm{mcm}(1, 2, ..., M)$$

che è assurdo. □

A breve risulterà molto utile il contenuto della seguente

Osservazione 7. Siano r un intero positivo e p un numero primo che non divide r. Consideriamo il polinomio $x^r - 1$ su $\mathbb{Z}_p$, avente per radici, nel suo campo di riducibilità completa, tutte le radici r-esime dell'unità, le quali, come abbiamo visto nella proposizione 1.2.8, formano un gruppo ciclico R_r di ordine r. Dunque esiste $\xi \in R_r$ di ordine r. Consideriamo il suo polinomio minimo $h(x)$ in $\mathbb{Z}_p$ e ne sia d il grado. Si ha $F = Z_p[x]/(h(x)) = \mathbb{Z}_p(\xi)$ e questo campo ha ordine p^d.
Notiamo che F, contenendo ξ, ne contiene anche tutte le potenze, e dunque contiene R_r, da cui F coincide con il campo di spezzamento di $x^r - 1$ su $\mathbb{Z}_p$.

Introduciamo ora la seguente terminologia.

Definizione 2.1. *Sia p un numero primo e r un intero positivo. Siano dati un polinomio $f(x) \in \mathbb{Z}[x]$ e un intero positivo m. Si dice che m è* ***introspettivo*** *per f(x) modulo p, relativamente a r, se si ha*

$$(f(x))^m = f(x^m) \quad in\ \mathbb{Z}_p[x]/(x^r - 1).$$

Lemma 2.0.5. *Se m e m' sono numeri introspettivi per un polinomio f(x) modulo p, relativamente ad r, allora il loro prodotto $m \cdot m'$ è ancora introspettivo rispetto allo stesso polinomio.*

Dimostrazione. Poiché m è introspettivo per $f(x)$, si ha:

$$[f(x)]^{m \cdot m'} = [f(x^m)]^{m'} \quad in\ \mathbb{Z}_p[x]/(x^r - 1).$$

Inoltre, poiché anche m' è introspettivo per $f(x)$, sostituendo x con x^m nell'equazione di introspezione per m', otteniamo:

$$[f(x^m)]^{m'} = f(x^{m \cdot m'})\ in\ \mathbb{Z}_p[x]/(x^{mr} - 1),$$

dove il secondo membro risulta essere uguale a $f(x^{m \cdot m'})$ in $\mathbb{Z}_p[x]/(x^r - 1)$ poiché $x^r - 1$ divide $x^{m \cdot r} - 1$.
Allora, confrontando le due equazioni di cui sopra, abbiamo:

$$[f(x)]^{m \cdot m'} = f(x^{m \cdot m'}) \quad in\ \mathbb{Z}_p[x]/(x^r - 1),$$

da cui la tesi. □

Lemma 2.0.6. *Se un numero è introspettivo per due polinomi f(x) e g(x) modulo p, relativamente ad r, allora esso è introspettivo per la loro somma e il loro prodotto.*

Dimostrazione. Dimostriamo che, dato un numero m, l'insieme dei polinomi per i quali m è introspettivo è chiuso rispetto alla moltiplicazione.
Si ha infatti:

$$[f(x) \cdot g(x)]^m = [f(x)]^m \cdot [g(x)]^m = f(x^m) \cdot g(x^m) \quad in\ \mathbb{Z}_p[x]/(x^r - 1).$$

Analogamente, l'insieme dei polinomi per cui m è introspettivo è chiuso rispetto alla somma. □

Abbiamo ora le basi per dimostrare il seguente risultato, fondamentale per l'algoritmo AKS.

Teorema 2.0.7. *Sia $n > 1$ un intero che non sia potenza di un primo con esponente maggiore di 1. Sia $r < n$ un intero positivo primo con n tale che*

$$r \leq [16 \log_2^5 n] + 1, \quad \mathrm{G_{SS}}(r, n) > 4 \log_2^2 n.$$

Supponendo che n non abbia fattori primi minori o uguali a r, allora n è primo se e solo se $\forall\ a = 1, ..., [2\sqrt{\varphi(r)}\log_2 n]$ vale la (2.2), ovvero

$$(x + a)^n = x^n + a \quad in \quad \mathbb{Z}_n[x]/(x^r - 1).$$

Dimostrazione. Per la proposizione 2.0.3 sappiamo che, se n è primo, vale la (2.1) e quindi la (2.4) $\forall a$. Pertanto, occorre soltanto provare che la condizione data dal teorema è sufficiente a garantire che n sia primo.
Procediamo per assurdo. Supponiamo, quindi, che esista p divisore primo di n. Allora $p < \frac{n}{2}$ e $p > r$, per cui $MCD(p, r) = 1$. Poniamo

$$k := [2\sqrt{\varphi(r)}\ \log_2 n].$$

Si ha

$$k < r - 1 \tag{2.6}$$

Infatti, se fosse $r - 1 \leq k$, si avrebbe $r - 1 \leq 2\sqrt{\varphi(r)}\ \log_2 n \leq 2\sqrt{r-1}\ \ \log_2 n$, da cui, dividendo per $\sqrt{r-1}$, $\sqrt{r-1} \leq 2\ \log_2\ n$. Elevando al quadrato, si ottiene quindi $r-1 \leq 4\ \log_2^2\ n \leq \mathrm{G_{SS}}(n, r)$, il che è assurdo. Per ipotesi la (2.4) è verificata $\forall\ a = 1, ..., l$. Essendo p un divisore di n, allora

$$(x + a)^n = x^n + a \quad \text{in} \quad \mathbb{Z}_p[x]/(x^r - 1) \quad \forall\ a = 1, ..., k. \tag{2.7}$$

Si deduce, quindi, che n è introspettivo modulo p e relativamente ad r per i polinomi $x + 1$, ... , $x + k$. Consideriamo adesso i seguenti insiemi:

$$I := \{n^i \cdot p^j | i, j \geq 0\}, \qquad P := \{\prod_{a=1}^{k} (x + a)^{i_a} \mid i_a \geq 0\}.$$

Si noti che ciascun elemento di I è introspettivo per ogni polinomio di P. Definiamo ora l'insieme

$$G := \{[x]_r | x \in I\}.$$

Per ipotesi $MCD(n, r) = 1$ e, per quanto visto sopra, anche $MCD(p, r) = 1$. G risulta dunque essere il sottogruppo di $\mathbb{Z}_r^*$ generato da n e p modulo r, ovvero G contiene le classi modulo r degli elementi di I. Ponendo $|G| = t$, si ha

$$\varphi(r) \geq t \geq \mathrm{G_{SS}}(n, r) > 4 \log_2^2 n, \quad t \geq \mathrm{G_{SS}}(p, r). \tag{2.8}$$

Consideriamo ora il campo $F = \mathbb{Z}_p(\xi) = \mathbb{Z}_p[x]/(h(x))$ come descritto nell'osservazione 7 e indichiamo con Q il sottogruppo del gruppo moltiplicativo F^* generato dalle classi dei polinomi dell'insieme P. Sia $q = |Q|$.
Q è un gruppo ciclico, pertanto esisterà un suo generatore γ che è la classe modulo p e modulo $h(x)$ di un polinomio $G(x) \in \mathbb{Z}[x]$. È utile notare che

$$m, m' \textit{ introspettivi per } G(x),\ m \equiv m' \pmod{r} \Rightarrow m \equiv m' \pmod{q}. \tag{2.9}$$

Infatti, se $m' \equiv m \pmod{r}$, si ha $m' = m + rv$, per cui

$$(G(x))^{m'} = (G(x))^{m+rv} = G(x^{m+rv}) \quad in\ \mathbb{Z}_p[x]/(x^r - 1),$$

ossia

$$(G(x))^m (G(x))^{rv} = (G(x))^m = G(x^m) \quad in\ \mathbb{Z}_p[x]/(x^r - 1).$$

Poiché $h(x) \mid x^r - 1$, vale anche

$$(G(x))^m (G(x))^{rv} = (G(x))^m \quad in\ F$$

e questo ci fa comprendere che $\gamma^{rv} = 1$ in F^*, per cui $q \mid rv$, ovvero $q \mid m$ - m' .
Osserviamo che q verifica contemporaneamente le seguenti disuglianze:

$$q \geq \binom{t+k-2}{t-1} \tag{2.10}$$

$$q < \frac{1}{4} n^{2\sqrt{t}}. \tag{2.11}$$

Per dimostrare la (2.10), consideriamo l'insieme P' dei polinomi in P della forma $\prod_{a=1}^{k} (x + a)^{i_a}$ e il cui grado sia $\sum_{a=1}^{k} i_a \leq t - 1$. Si dimostra per induzione che il loro

numero è esattamente $\binom{t+k-2}{t-1}$ (cfr [2], esercizio A1.24 pag 65). In più, ora proveremo che le loro classi in F^* sono distinte, affermazione dalla quale segue direttamente la tesi. Notiamo innanzitutto che i polinomi in P' sono tutti distinti in $\mathbb{Z}_p[x]$, poiché $k < p$, altrimenti dalla (2.6) avremmo $p \leq k < r-1$, contro l'ipotesi $p > r$.
Per assurdo, supponiamo che in F^* le classi di $f(x)$ e $g(x)$ polinomi in P' non siano distinte. Poniamo $l(x) = f(x) - g(x)$ e denotiamo con $\bar{l}(x)$ la classe di l(x) in $\mathbb{Z}_p[x]$. Allora il polinomio $\bar{l}(x)$ è divisibile per $h(x)$ e dunque ha come radice la radice primitiva r-esima ξ dell'unità, ovvero

$$\bar{l}(\xi) = 0.$$

Il fatto che tutti i numeri in I siano introspettivi modulo p relativamente ad r sia per $f(x)$ sia per $g(x)$ implica che tali numeri siano introspettivi anche per $l(x)$, ossia, $\forall\ i, j$ interi non negativi,

$$(\bar{l}(x))^{n^i \cdot p^j} = \bar{l}(x^{n^i \cdot p^j}) \quad in \quad \mathbb{Z}_p[x]/(x^r - 1).$$

Poiché $h(x) | x^r - 1$, vale anche

$$(\bar{l}(x))^{n^i \cdot p^j} = \bar{l}(x^{n^i \cdot p^j}) \quad in \quad F = \mathbb{Z}_p[x]/(h(x))$$

$\forall\ i, j$ interi non negativi. Ne segue che, $\forall i, j$ interi non negativi,

$$\bar{l}(\xi^{n^i \cdot p^j}) = \bar{l}(\xi)^{n^i \cdot p^j} = 0.$$

Le classi modulo r dei numeri $n^i \cdot p^j$ al variare di i e j sono tante quante gli elementi di G, ovvero t, pertanto $\bar{l}(x) \in \mathbb{Z}_p[x]$ ha t radici distinte ξ, ξ^2,..., ξ^n. Poiché $\bar{l}(x)$ ha grado minore di t, esso è nullo in $\mathbb{Z}_p[x]$. Dunque $f(x)$ e $g(x)$ sono uguali modulo p, da cui $f(x) = g(x)$. Si conclude così la prova di (2.10).
Dimostriamo adesso la proprietà (2.11). Procediamo per assurdo, supponendo quindi che

$$q \geq \frac{1}{4} n^{2\sqrt{t}}. \tag{2.12}$$

Prendiamo in considerazione gli elementi del tipo $n^i\ p^j$ in I con $0 \leq i\ \leq \sqrt{t}$ e $0 \leq j \leq \sqrt{t}$, che sono di numero $([\sqrt{t}] + 1)^2 > t$. Dunque le classi degli elementi siffatti non sono tutte distinte in G, per cui $\exists\ i, i', j, j'$ con $0 \leq i, i', j, j' \leq \sqrt{t}$, $i \neq i'$ e $j \neq j'$ tali che

$$n^i p^j \equiv n^{i'} p^{j'} \pmod{r}. \tag{2.13}$$

$n^i p^j$ e $n^{i'} p^{j'}$ sono introspettivi per i polinomi di P, pertanto lo sono anche per $G(x) \in \mathbb{Z}[x]$ la cui classe in F genera Q e quindi, per la (2.9), vale

$$n^i p^j \equiv n^{i'} p^{j'} \pmod{q}. \tag{2.14}$$

Supponendo $i \geq i'$ e $j \geq j'$, abbiamo $i,\ j \leq \sqrt{t}$, per cui $n^i \leq n^{\sqrt{t}}$ e $p^j \leq (\frac{n}{2})^{\sqrt{t}}$. Allora, essendo $t > 4$ per la (2.8), si ha $1 \ \leq n^i p^j \leq 1/4 n^{2\sqrt{t}} \leq q$ e $1 \ \leq n^{i'} p^{j'} \leq q$. Dalla (2.13) deduciamo che $n^i p^j = n^{i'} p^{j'}$, da cui $i = i'$ e $j = j'$, il che è assurdo. Consideriamo ora il caso $i \leq i'$ e $j \leq j'$. Abbiamo

$$n^{i-i'} \equiv p^{j'-j} \pmod{q},$$

da cui, ragionando come sopra, $n^{i-i'} = p^{j'-j}$. Se n non fosse primo, sarebbe potenza di p e si otterrebbe un assurdo.

Alla luce di questi risultati, possiamo concludere la dimostrazione del teorema. Per la (2.8) vale $t - 1 \geq s$ e, per la (2.10), si ha

$$q \geq \binom{t+k-1}{s}, \quad \text{dove } s = [2\sqrt{t}\,\log_2 n].$$

Notiamo che $k \geq 2\sqrt{\varphi(r)}\,\log_2 n \ \geq 2\sqrt{t}\,\log_2 n$, da cui $k \geq s$ e quindi

$$q \geq \binom{2s-1}{s}.$$

Per $s \geq 3$ si ha

$$\binom{2s-1}{s} = \frac{2s-1}{s-1} \cdot \frac{2s-2}{s-2} \cdot \ldots \cdot \frac{s+2}{2} \cdot (s+1) \geq 2^{s-1}$$

e, di conseguenza,

$$q \geq 2^{s-1} \geq 2^{2\sqrt{t}\,\log_2 n - 2} = \frac{1}{4} n^{2\sqrt{t}},$$

che è un assurdo poichè abbiamo provato che vale la (2.11). Questa contraddizione deriva dall'aver assunto n non primo, per cui n è primo. □

Capitolo 3

La correttezza e la complessità di AKS

3.1 L'algoritmo AKS e la sua correttezza

Descriviamo ora l'algoritmo AKS nella maniera in cui esso è stato presentato dai matematici Agrawal, Kayal e Saxena.
Dato $n > 1$ si vuole stabilire se n è composto. L'algoritmo procede nel seguente modo:

1. Se ($n = m^k$ per m, k interi e $k > 1$), allora n è COMPOSTO;
2. si determini il più piccolo intero r tale che $\mathrm{G_{SS}}(r, n) > 4\log_2^2 n$;
3. se $1 < MCD(a, n) < n$ per qualche $a \leq r$, n è COMPOSTO;
4. se $n \leq r$, n è PRIMO;
5. se $(x + a)^n \neq x^n + a$ in $\mathbb{Z}_n[x]/(x^r - 1)$ $\forall\ a = 1, ..., \lfloor 2\sqrt{\varphi(r)}\ \log_2 n \rfloor$, rispondi n COMPOSTO;
6. se in 5 vale l'uguaglianza, allora n è PRIMO.

Commentiamo ora tale algoritmo.
Nel caso (1), essendo n potenza di un intero con esponente maggiore di 1, ovviamente n non è primo.

Il caso (2) è giustificato dal lemma 2.0.4, nel quale si è provato che esiste un $r \leq [16\log_2^5 n]$ tale che $G_{SS}(r, n) > 4 \log_2^2 n$.
Il punto (3) consiste nel calcolare r volte un massimo comun divisore: se uno di questi è strettamente maggiore di 1 e strettamente minore di n, ovviamente n è composto.
Nel punto (4), $n \leq r$ è necessariamente primo perché, se fosse composto, il punto precedente lo avrebbe già riconosciuto come tale: infatti, è sufficiente prendere come a un fattore non banale di n e, poiché $n \leq r$, $a \leq r$ e, di conseguenza, si avrebbe $1 < MCD(a, n) = a < n$.
Infine il punto (5) deriva dal teorema 2.0.7
Dalle considerazioni precedenti, è facile dedurre la correttezza dell'algoritmo AKS.

3.2 La complessità computazionale di AKS

Analizziamo in questa sezione la complessità computazionale dell'algoritmo AKS, dimostrando che quest'ultimo possiede tempo polinomiale.
Enunciamo un risultato concernente la complessità computazionale del primo punto dell'algoritmo. Abbiamo infatti la seguente

Proposizione 3.2.1. *Sia $n > 1$ intero positivo. Allora esiste un algoritmo polinomiale di complessità $O(\log^4 n)$ che riconosce se esistono interi positivi m, k con $k > 1$ tali che $n = m^k$*

Dimostrazione. Se esistono m, k, $k > 1$ interi positivi , tali che $n = m^k$, si ha $k = \log_m n < \log_2 n$. Di conseguenza, ponendo $h = [\log_2 n]$ si ha $2 \leq k \leq h$. Dobbiamo allora verificare che, $\forall$ k siffatto, il polinomio $f_k(x) = x^k - n$ ha una radice intera positiva.
Per provare che tale radice esiste ed è unica, si noti che il polinomio $f_k(x)$ è strettamente crescente nell'intervallo $[0,\ n]$ e $f_k(0) = -n < 0$, mentre $f_k(n) = n^k - n > 0$, da cui in $[0,n]$ si ha una e una sola radice.
Dobbiamo verificare che essa è intera. Valutiamo il segno di $f_k(\frac{n}{2})$: dobbiamo calcolare $n^k - 2^k \cdot n$, avente complessità $O(k \log\ n) = O(\log^2 n)$. Se fosse $f_k(\frac{n}{2}) = 0$, il che non si verifica nella pratica, avremmo finito. Supponiamo che $f_k(\frac{n}{2})$ sia positivo. Valutiamo allora $f_k(x)$ in $n/4$, ovvero nel punto medio dell'intervallo $[0,\ n/2]$. Occorre perciò

calcolare $n^k - 4^k \cdot n$, il che ha di nuovo complessità $O(\log^2 n)$. Se $f_k(n/4) = 0$, abbiamo concluso. In caso contrario, iteriamo il procedimento fin qui descritto, supponendo di non trovare mai la radice k-esima di n in [0,n] nelle valutazioni successive di $f_k(x)$ nei punti medi di successivi intervalli ai cui estremi $f_k(x)$ assume valori di segni opposti. Dopo $h+1$ passi, otterremo una successione finita x_0,...,x_{h+1} di punti di [0, n] tali che:

- $\forall\ i = 1,...,\ h+1$, $f_k(x)$ assume valori di segni opposti in x_{i-1} e in x_i;
- $|x_i - x_{i-1}| = n/2^i$.

La costruzione di questa successione ha costo computazionale polinomiale, pari a $O(h\log^2 n) = O(\log^3 n)$. In particolare avremo localizzato la radice k-esima di n in [0, n], che deve appartenere all'intervallo $[x_h, x_{h+1}]$, avente lunghezza $n/2^{h+1} < 1$ e, per questo motivo, contenente al più un intero. Se in tale intervallo non ci sono interi, allora n non ha radice k-esima intera. Al contrario, se nell'intervallo è presente l'intero m, calcolando m^k si verifica se la radice k-esima di n è o non è un intero. Il procedimento descritto va ripetuto $\forall\ k$ tale che $2 \leq k \leq h$, per cui la complessità di tale algoritmo è $O(h\log^3 n)$ $= O(\log^4 n)$, come volevasi dimostrare. □

Siamo ora in grado di dimostrare il seguente

Teorema 3.2.2. *Sia n un intero maggiore di 1. Allora l'algoritmo AKS ha tempo polinomiale.*

Dimostrazione. Per provare che AKS applicato ad n ha tempo polinomiale, è sufficiente dimostrare che ciascun punto dell'algoritmo è polinomiale.
Andiamo allora ad analizzare la complessità di ogni passo.

- Il punto (1) ha complessità $O(\log^4 n)$ per la Proposizione 3.2.1
- Nel passo (2) si deve controllare che $n^k \neq 1 \pmod r$ per ogni $k \leq 4\log^2 n$. Questo richiede il calcolo di, al più, $O(\log^2 n)$ potenze modulo r, che ha complessità $O(\log^2 n\ \log^3 r)$ (cfr [2], paragrafo 3.3.1). Dal lemma 2.0.4 sappiamo che si devono testare al massimo $[16\log^5 n + 1]$ interi r, dunque, in definitiva, la complessità totale di questo punto è $O(\log^{7+\epsilon} n)$, con $\epsilon > 0$.

- Il punto (3) consiste nel calcolare r volte un massimo comun divisore, ognuno dei quali richiede un costo computazione pari a $O(\log^3 n)$ per la proposizione 1.3.2. La complessità totale di questo passo risulterà allora $O(r \cdot \log^3 n)$, che, sempre utilizzando il lemma 2.0.4, sarà uguale a $O(\log^8 n)$.

- Il passo (4) è a complessità al più $O(\log\ n)$, ma si nota che in fase di implementazione i passi (3) e (4) possono essere uniti.

- Il punto (5) rappresenta il calcolo più oneroso dell'intero algoritmo. Bisogna infatti controllare $[2\sqrt{\varphi(r)}\ \log\ n]$ equazioni, ovvero, per ognuna di esse, occorre eseguire $O(\log\ n)$ moltiplicazioni tra polinomi di grado r a coefficienti di $O(\log\ n)$ bit. Complessivamente si hanno $O(r^2 \log^3 n)$ (cfr [2], es.A6.26 pag.310) operazioni per singola equazione, per cui la complessità totale di questo passo risulta essere

$$O(r^2\sqrt{\varphi(r)}\log^3 n) = O(r^{5/2}\log^3 n) = O(\log^{15.5} n).$$

  dove l'ultima uguaglianza deriva sempre dal lemma 2.0.4.

La complessità di quest'ultimo passo domina la complessità di tutti gli altri punti e, pertanto, la complessità dell'intero algoritmo AKS è $O(\log^{15.5} n)$. □

In [1] è stata data una stima migliore di tale complessita, pari a $O(\log^{\frac{21}{2}+\epsilon} n)$, con $\epsilon > 0$.
Negli anni successivi al 2002, diversi matematici, tra i quali ricordiamo Hendrick Lenstra, Daniel Bernstein e Carl Pomerance, hanno messo in atto tecniche per migliorare il tempo di esecuzione dell'algoritmo, affinché esso sia utilizzabile nella pratica, portando la complessità ad un $O(\log^{6+\epsilon} n)$, con $\epsilon > 0$, sotto l'ipotesi che sia vera una congettura sulla distribuzione dei numeri primi di Sophie-Germain.

Bibliografia

[1] M. Agrawal, N. Kayal, N. Saxena. *PRIMES is in P*. Ann. of Math. (2), **160** (2002), n. 2, 781-793

[2] M. W. Baldoni, C. Ciliberto, G. M. Piacentini Cattaneo. *Aritmetica, crittografia e codici*. Springer, Milano, 2006

[3] M. Morigi, *Note del corso di Algoritmi della Teoria dei Numeri e Crittografia*, Bologna, 2006

[4] A. Vistoli, *Note di Algebra*, Bologna, 1994

[5] R. Betti, *Numeri primi e complessità: l'algoritmo AKS* in http://matematica.unibocconi.it/articoli/numeri-primi-e-complessitàlalgoritmo-aks