Qualche rudimento sui campi finiti per il corso di Teoria dell'Informazione e Complessità

Agostino Dovier

December 1, 2015

1 Promemoria

- $(A, \circ)$ è *semigruppo* se $\circ : A \times A \longrightarrow A$ è associativo, ovvero, per ogni $a, b, c \in A$ vale che:

$$(a \circ b) \circ c = a \circ (b \circ c)$$

- $(A, \circ)$ è *monoide* se è semigruppo e esiste l'elemento neutro 0 t.c. per ogni $a \in A$:

$$a \circ 0 = 0 \circ a = a$$

- $(A, \circ)$ è *gruppo* se è un monoide e per ogni $a \in A$ esiste $b \in A$ (opposto di a) tale che:

$$a \circ b = b \circ a = 0$$

- $(A, \circ)$ è *gruppo abeliano* se è un gruppo e per ogni $a, b \in A$ vale che:

$$a \circ b = b \circ a$$

- $(A, +, \cdot)$ è *anello* se:
  - $(A, +)$ è gruppo abeliano (sia 0 l'elemento neutro di '+'),
  - $(A, \cdot)$ è semigruppo (sia 1 l'elemento neutro di '$\cdot$'),
  - vale la proprietà di *distributività*, ovvero: per ogni $a, b, c \in A$:

$$\begin{array}{l} a \cdot (b + c) = (a \cdot b) + (a \cdot c) \quad \text{e} \\ (b + c) \cdot a = (b \cdot a) + (c \cdot a) \end{array}$$

- $(A, +, \cdot)$ è *anello commutativo* se è anello e per ogni $a, b \in A$ vale che:

$$a \cdot b = b \cdot a$$

- $(A, +, \cdot)$ è *campo* se:
  - $(A, +, \cdot)$ è anello commutativo con 0 elemento neutro di '+', e
  - ogni elemento non nullo (diverso da 0) è inverbile rispetto a '$\cdot$', ovvero, per ogni $a \in A$, $a \neq 0$, esiste $b \in A$ tale che $ab = 1$.

  Al solito, il simbolo '$\cdot$' viene omesso, quando chiaro dal contesto.

2 Campi Finiti

$(A, +, \cdot)$ è *campo finito* se è un campo e A contiene un numero finito di elementi. Galois ci insegna che se $(A, +, \cdot)$ è campo finito, allora $|A| = p^r$ elementi, ove p è un numero primo e $r \in \mathbb{N}$.

In particolare $\mathbb{Z}_p = (\{0, 1, \ldots, p-1\}, +, \cdot)$ con l'algebra in modulo p, se p è primo, è un campo finito (si verifichino le definizioni per $\mathbb{Z}_2$).[1] Se p non è primo, la proprietà non vale. Si pensi ad esempio a $\mathbb{Z}_4$:

$$2 \cdot 1 = 2, 2 \cdot 2 (= 4) = 0, 2 \cdot 3 (= 6) = 2$$

Gli elementi 2 e 3 non sono invertibili rispetto a '$\cdot$'.

Galois ci dice anche che per ogni numero primo p ed intero r esiste un campo finito con p^r elementi, unico a meno di isomorfismi, e si indica con $GF(p^r)$.[2]

La tecnica per operare in $GF(p^r)$ è quella di considerare i suoi elementi come polinomi di grado $0, 1, \ldots, r-1$ a coefficienti in $\mathbb{Z}_p$. Ad esempio, in $GF(2^2)$ si considerino:

$$\begin{array}{ccc} & & 0 \\ & & 1 \\ x & & \\ x & + & 1 \end{array}$$

I polinomi sono elementi di una struttura A per cui esiste l'operazione di somma (associativa, commutativa, con elemento neutro 0, con opposto: gruppo abeliano). Moltiplicando due polinomi invece si può uscire da A (ad esempio $x \cdot x = x^2$). Se prendiamo il modulo della divisione del risultato con $x^2 + x + 1$ siamo garantiti di rimanere in A. $x^2 + x + 1$ è un *polinomio irriducibile* di grado r ($r = 2$ in questo caso). Se consideriamo dunque A con l'operazione di somma usuale tra polinomi, ma usando i coefficienti con le regole di $\mathbb{Z}_p$, e il prodotto usando prima le regole usuali del prodotto di polinomi e del prodotto in $\mathbb{Z}_p$ e poi prendendo il modulo della divisione con un polinomio irriducibile di grado r, otteniamo un campo finito.

Nell'esempio di $GF(2^2)$, l'inverso di x è $x+1$ (e viceversa). Infatti $x(x+1) = x^2 + x$. Anzichè dividere possiamo sommare $x^2 + x + 1$ che è congruo a 0. Si ottiene (ricordo che, in $\mathbb{Z}_2$, $1 + 1 = 0$):

$$x^2 + x + x^2 + x + 1 = x^2(1 + 1) + x(1 + 1) + 1 = x^2 0 + x0 + 1 = 1$$

Pertanto $GF(p^r)$ è isomorfo a una struttura i cui elementi sono i polinomi a coefficienti in p di grado minore a r (sono in effetti p^r) le cui operazioni sono la somma e il prodotto, con le regole standard dei polinomi per raggruppare monomi dello stesso grado, con le regole di $\mathbb{Z}_p$ per le somme locali, e con una finale operazione in modulo, usando un polinomio irriducibile di grado r (fissato).

$GF(2^3)$, con polinomio irriducibile $x^3 + x + 1$ può essere sintetizzato dalle tabelle in Figura 1.

Ad esempio, il calcolo di $x^2(x + 1)$ avviene come segue:

$$x^2(x + 1) = x^3 + x^2 = x^3 + x^2 + x^3 + x + 1 = x^2 + x + 1$$

mentre

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

Si noti che nel calcolare il quadrato di $(x^2 + x + 1)$ i "doppi prodotti" non sono stati dimenticati, ma, al solito, in $\mathbb{Z}_p$, 2 è congruo a 0.

Osserviamo sull'esempio una proprietà che poi riconosceremo come teorema. Prediamo un elemento del campo diverso da 0 e 1 e calcoliamo le sue potenze. Ad esempio,

$$x, x^2, x^3 = x + 1, x^4 = x^2 + x, x^5 = x^2 + x + 1, x^6 = x^2 + 1, x^7 = 1$$

Partendo da un altro elemento α diverso da 0 e 1, avremmo ottenuto un risultato analogo, ovvero:

- $\alpha^1, \alpha^2, \cdots, \alpha^7$ enumera tutti gli elementi del campo (salvo lo 0) e

[1] Si eseguano le operazioni in $\mathbb{Z}$ e si prenda poi il resto della divisione del risultato con p.

[2] GF sta per Galois Field, ovviamente. Ometto qui la nozione precisa di isomorfismo; ci basta pensare ad un isomorfismo come una rinomina consistente degli elementi del campo.

$+$	0	1	x	$x+1$	x^2	x^2+1	x^2+x	x^2+x+1
0	0							
1	1	0						
x	x	$x+1$	0					
$x+1$	$x+1$	x	1	0				
x^2	x^2	x^2+1	x^2+x	x^2+x+1	0			
x^2+1	x^2+1	x^2	x^2+x+1	x^2+x	1	0		
x^2+x	x^2+x	x^2+x+1	x^2	x^2+1	x	$x+1$	0	
x^2+x+1	x^2+x+1	x^2+x	x^2+1	x^2	$x+1$	x	1	0

$\cdot$	0	1	x	$x+1$	x^2	x^2+1	x^2+x	x^2+x+1
0	0							
1	0	1						
x	0	x	x^2					
$x+1$	0	$x+1$	x^2+x	x^2+1				
x^2	0	x^2	$x+1$	x^2+x+1	x^2+x			
x^2+1	0	x^2+1	1	x^2	x	x^2+x+1		
x^2+x	0	x^2+x	x^2+x+1	1	x^2+1	$x+1$	x	
x^2+x+1	0	x^2+x+1	x^2+1	x	1	x^2+x	x^2	$x+1$

Figure 1: Le tabelle delle operazioni '+' e '.' in $GF(2^3)$ usando polinomio irriducibile x^3+x+1

- $\alpha^7 = 1$.

La teoria dei campi finiti ci dice infatti che gli elementi di $GF(p^r)$ sono *tutte e sole le radici di:*

$$X^{p^r} - X = 0$$

Dunque $X = 0$, più le radici $p^r - 1$ (anche dette *primitive*) dell'unità. Nell'esempio sopra: $X = 1, X = x, \ldots, X = x^2 + x + 1$.[3]

Sottolineiamo inoltre un'altra proprietà particolarmente interessante per noi informatici. Gli elementi di $GF(2^r)$ possono essere rappresentati univocamente con r-uple di 0 e 1. Ad esempio, gli elementi di $GF(2^2)$ sopra sarebbero $(0,0),(0,1),(1,0),(1,1)$. L'elemento i-esimo della r-upla è il coefficiente a_i del generico polinomio

$$a_{r-1}x^{r-1} + \cdots a_1 x + a_0$$

Pertanto, visto che $GF(2^1), GF(2^2), GF(2^3), \ldots$ sono tutti campi finiti possiamo ragionare in termini di r-uple su campi ben popolati (ad esempio $GF(2^{256})$) e rappresentare gli elementi usando r-uple di bits.

3 Alcuni risultati per i codici ciclici

Un codice $\mathcal{C} \subseteq \mathcal{X}^n$ è un insieme di n-uple di elementi in $\mathcal{X}$. Nei codici *algebrici* $\mathcal{X}$ deve essere un campo finito, diciamo $GF(q)$ con $q = p^r$, dunque ogni elemento di $\mathcal{X}$:

- Nel caso $|\mathcal{X}| = p$, con p numero primo, può essere assimilato ad un elemento di $\mathbb{Z}_p$
- Nel caso più generale in cui $|\mathcal{X}| = p^r$, con $r > 1$ può essere assimilato ad un polinomio con variabile a di grado minore di r a coefficienti in $\mathbb{Z}_p$ (o, equivalentemente, ad una r-upla di elementi di $\mathbb{Z}_p$)

Per fissare le idee, qualora gli elementi di $\mathcal{X}$ siano interpretati come polinomio, scegliamo a come variabile per i polinomi (ad esempio $a^2 + a + 1 \in GF(2^3)$).

Consideriamo dunque

$$\vec{c} = \langle c_0, c_1, \ldots, c_{n-1} \rangle \in \mathcal{C}$$

Aggiungendo un ulteriore livello di astrazione, interpretiamo anche $\vec{c}$ come un polinomio, questa volta nella variabile x:

$$c(x) = c_0 + c_1 x + \cdots + c_{n-1}x^{n-1}$$

[3] Si noti l'uso della X maiuscola come incognita del sistema e della x minuscola come monomio fondamentale per la costruzione dei polinomi.

Sia R_n la struttura algebrica i cui elementi sono i polinomi di grado minore di n a coefficienti in $GF(q)$, e le operazioni si fanno in modulo $x^n - 1$ (polinomio, si noti, non irriducibile per $n > 1$). La struttura è un anello commutativo.

Un codice $\mathcal{C} \subseteq \mathcal{X}^n$ è un *codice ciclico* se è un codice lineare e ogni permutazione di una parola di codice è ancora parola di codice. Nell'interpretazione di $\mathcal{C}$ con i polinomi in x, si osservi che

$$xc(x) = c_0x + c_1x^2 + \cdots + c_{n-1}x^n \underline{\mathsf{mod}}\, (x^n - 1) = c_{n-1} + c_0x + c_1x^2 + \cdots + c_{n-2}x^{n-1}$$

ovvero corrisponde ad uno shift ciclico della parola di codice.

[...]

Abbiamo visto che la progettazione dei codici ciclici è basata sul calcolo dei fattori di $x^n - 1$ per determinare i polinomi $g(x)$ e $h(x)$ usati nella matrice generatrice e di controllo, rispettivamente.

Per far ciò facciamo in modo che anche n ci permetta di ragionare nei campi finiti. Prendiamo dunque $n = p^m - 1$ per qualche primo p e intero m (ovvero tutti gli elementi di $GF(p^m)$ meno lo zero).

Poichè gli elementi non nulli del campo finito $GF(p^m)$ sono tutte radici primitive dell'unità sembrerebbe che essi siano gli elementi che mi permettono di fattorizzare $x^{p^m} - 1$, ovvero, se α è elemento del campo finito, non nullo, non unitario, allora:

$$x^{p^m} - 1 = (x - \alpha)(x - \alpha^2)(x - \alpha^3)\cdots(x - \alpha^{p^m-2})(x - 1)$$

($\alpha^{p^m-1} = 1$). Ma le cose sono più complesse.

Ragioniamo in $GF(2^3)$. Dobbiamo scomporre $x^7 + 1$. Abbiamo che:

$$x^7 + 1 = (x + 1)(x^3 + x + 1)(x^3 + x^2 + 1)$$

ma i polinomi sulla destra sono tutti irriducibili (proprietà verificabile con pochi calcoli). Come sono collegati questi polinomi agli elementi non nulli $1, a, a + 1, a^2, \ldots, a^2 + a + 1$ di $GF(2^3)$?

$x + 1$ è proprio il polinomio $x - 1$ ($-1 = 1$ in $\mathbb{Z}_2$).

$x^3 + x + 1$ Si osservi che a è una sua soluzione. Infatti $a^3 + a + 1 = a^3 + a + 1 + a^3 + a + 1 = 0$ (ricordiamo che siamo in modulo $a^3 + a + 1$). Ma anche a^2 è una soluzione, infatti: $a^6 + a^2 + 1 = a^3a^3 + a^2 + 1 = (a + 1)(a + 1) + a^2 + 1 = a^2 + 1 + a^2 + 1 = 0$ (ricordiamo che $a^3 = a^3 + a^3 + a + 1 = a + 1$)). Cosí come a^4: $a^12 + a^4 + 1 = a^7a^5 + a^3a + 1 = 1a^3a^2 + (a + 1)a + 1 = (a + 1)a^2 + a^2 + a + 1 = a^3 + a^2 + a^2 + a + 1 = a^3 + a + 1 = 0$.

$x^3 + x^2 + 1$ Similmente, si verifica che questo polinomio ha come soluzioni $a^3, a^6, a^{12} = a^5$.

Gli elementi del campo permettono di aiutare a scomporre il polinomio $x^n - 1$, ma non di arrivare a polinomi di grado minimo (si pensi ad un caso analogo: scomporre $x^2 + 1$ in $\mathbb{R}$ non è possibile, anche se ammette le radici immaginarie $\pm i$).

Si osserva (è in realtà un teorema!) che se β è radice di un polinomio, allora anche $\beta^{p^1}, \beta^{p^2}, \beta^{p^3}, \beta^{p^4}, \ldots$ lo sono (ovviamente vi è poi ciclicità dei valori).

Dunque, partendo da $GF(2^3)$ avremo che $\alpha, \alpha^2, \alpha^4, \alpha^8 = \alpha$ sono radici degli stessi polimoni. Dunque $a, a^2, a^4 = a^2 + a$ saranno sempre radici (coniugate) degli stessi polinomio; così come $a + 1, (a + 1)^2 = a^2 + 1, (a + 1)^4 = a^2 + a + 1, (a + 1)^8 = a + 1$ sono radici degli stessi polinomi.

Sulla base di queste (e altre) osservazioni, si possono identificare i polinomi minimi che ci servono per progettare i codici ciclici.

Ad esempio, sapendo che $a, a^2, a^2 + a$ sono le radici di un polinomio di grado 3, possiamo calcolarlo (cambio sempre - in + per comodità):

$$\begin{aligned}
(x + a)(x + a^2)(x + a^2 + a) &= x^3 + x^2(a + a^2 + a^2 + a) + x(a\,a^2 + a\,(a^2 + a) + a^2(a^2 + a)) + a\,a^2\,(a^2 + a) \\
&= x^3 + x(a^3 + a^3 + a^2 + a^4 + a^3) + a^5 + a^4 \\
&= x^3 + x(a^2 + (a + 1)a + a + 1) + (a + 1)a^2 + (a + 1)a \\
&= x^3 + x(a^2 + a^2 + a + a + 1) + a^3 + a^2 + a^2 + a \\
&= x^3 + x + a^3 + a + a^3 + a + 1 \\
&= x^3 + x + 1
\end{aligned}$$

Sia $\mathcal{C}$ un codice ciclico con generatore $g(x)$ sia α una radice di $g(x)$.

Se $c(x) \in \mathcal{C}$ allora, per definizione di generatore, si ha che $c(x) = f(x)g(x)$ per qualche polinomio f. Ma allora α è radice anche di $c(x)$.

Il seguente è un teorema: se $\alpha_1, \ldots, \alpha_t$ sono *tutte* le radici di $g(x)$ avremo che $c(x) \in \mathcal{C}$ se e solo se

$$\bigwedge_{i=1}^{t} c(\alpha_i) = 0$$

Dunque, per testare se $c = \langle c_0, \ldots, c_{n-1} \rangle$ è parola di codice, conocendo una radice α di $g(x)$ allora posso verificare se $c(\alpha) = 0$ (condizione necessaria), ovvero:

$$c_0 + c_1\alpha + c_2\alpha^2 + \cdots + c_{n-1}\alpha^{n-1}$$

O in altri termini, calcolo:

$$H = \begin{pmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{n-1} \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{pmatrix}$$

Abbiamo ritrovato il codice di Hamming. Se partiamo da un campo $GF(2^r)$ allora avremo che $\alpha, \alpha^2, \alpha^4$ etc. sono radici degli stessi polinomi. Non si guadagna nulla a verificare $c(\alpha^2), c(\alpha^4)$, etc.

Piuttosto α^3 è una radice indipendente. Dunque valutando $c(\alpha^3)$ si hanno nuove informazioni:

$$c_0 + c_1\alpha^3 + c_2(\alpha^3)^2 + \cdots + c_{n-1}(\alpha^3)^{n-1}$$

Dunque, se sia α che α^3 sono radici di $g(x)$ il seguente controllo sembra essere più efficace di quello sopra:

$$H = \begin{pmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{n-1} \\ 1 & \alpha^3 & (\alpha^3)^2 & \cdots & (\alpha^3)^{n-1} \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{pmatrix}$$

Abbiamo ritrovato il codice BCH.

Queste considerazioni conducono ad una controparte formale che è anche il teorema fondamentale per i codici ciclici: Sia $\alpha \in GF(p^r)$ e siano

$$\alpha^b \alpha^{b+1} \alpha^{b+2} \cdots \alpha^{b+\delta-2}$$

radici di $g(x)$. Allora con

$$H = \begin{pmatrix} 1 & \alpha^b & (\alpha^b)^2 & \cdots & (\alpha^b)^{n-1} \\ 1 & \alpha^{b+1} & (\alpha^{b+1})^2 & \cdots & (\alpha^{b+1})^{n-1} \\ & & & \vdots & \\ 1 & \alpha^{b+\delta-2} & (\alpha^{b+\delta-2})^2 & \cdots & (\alpha^{b+\delta-2})^{n-1} \end{pmatrix}$$

si controlla un codice ciclico con distanza minima almeno δ.

Alcune righe sono ridondanti (in $GF(2^r)$ tutte le righe pari, ad esempio).

4 Risultati preliminari per RSA

4.1 Euclide

Riportiamo qui il noto algoritmo di Euclide, che chiameremo MCD e, nella prossima sezione, la sua versione estesa EE che risulta utile per il calcolo delle inverse nei campi finiti.

Iniziamo con MCD, definito come (assumiamo $a > b$ in chiamata; se $a = b$ allora $\mathsf{MCD}(a, b) = a$; se $a < b$ li scambio):

```
MCD(a, b)
    if  b = 0
    return  a
    else  return  MCD(b, a mod b)
```

Correttezza: (1) Se $d|a$ e $d|b$ allora $a = kd$ e $b = hd$ per qualche coppia di interi h, k.[4] Dobbiamo mostrare che $d|b$ (già nelle ipotesi) e che $d|a \bmod b$. Ora, sappiamo che $a = bc + a \bmod b$ per qualche c, per definizione della divisione con resto. Pertanto

$$kd = hdc + a \bmod b \Rightarrow a \bmod b = d(k - hc)$$

dunque $d|a \bmod b$.
(2) Supponiamo ora $d|b$ e $d|a \bmod b$, dunque che $b = kd$ e $a \bmod b = hd$ per h, k opportuni. Poichè $a = bc + a \bmod b$, per qualche c, vale che

$$a = kdc + hd = d(kc + h)$$

e dunque $d|a$ ($d|b$ era un'ipotesi).

Complessità: Sia la successione di Fibonacci definita al solito come $F_0 = 1, F_1 = 1, F_{i+2} = F_{i+1} + F_i$. Vale il seguente risultato: se $\mathsf{MCD}(a, b)$ esegue almeno $k \geq 1$ chiamate ricorsive, allora $a \geq F_{k+1}$ e $b \geq F_k$.

Si dimostra per induzione su k. Con $k = 1$ significa che $a > b > 0$. Ovvero $b \geq 1 = F_1$ e $a \geq 2 = F_2$. Supponiamo ora che vengano eseguite $k + 1$ chiamate. Ciò significa che $\mathsf{MCD}(b, a \bmod b)$ necessita di k chiamate. Per ipotesi induttiva, $b \geq F_{k+1}$ e $a \bmod b \geq F_k$. Rimane da mostrare che $a \geq F_{k+2}$. Sappiamo che $a > b \geq F_k$. Ma $a = bh + a \bmod b$ per qualche h. Essendo che $a > b$, $h \geq 1$. Dunque $a \geq b + a \bmod b \geq F_{k+1} + F_k = F_{k+2}$.

Da questo risultato segue il Teorema di Lamè che dice che, se $a > b \geq 0$ e $b < F_k$, allora l'esecuzione di $\mathsf{MCD}(a, b)$ necessita di meno di k chiamate ricorsive, da cui, poichè F_k cresce esponenzialmente[5] si ha che il numero di chiamate (e di conseguenza la complessità dell'algoritmo di Euclide) è $O(\log b)$.

4.2 Euclide Esteso

Come accennato, l'algoritmo di Euclide può essere esteso per trovare i coefficienti interi x ed y tali che

$$ax + by = \mathsf{MCD}(a, b)$$

Denotiamo per semplicità con $d = \mathsf{MCD}(a, b)$. L'algoritmo seguente restituisce la tripla $\langle d, x, y\rangle$

```
EE(a, b)
    if  b = 0
    return  ⟨a, 1, 0⟩
    else  begin
              ⟨d', x', y'⟩ ← EE(b, a mod b);
              return  ⟨d', y', x' − (a div b)y'⟩
          end
```

Ad esempio, $\mathsf{MCD}(24, 30) = \langle 6, -6, 145\rangle$, $\mathsf{MCD}(144, 64) = \langle 16, 1, -16\rangle$.

Complessità. E' evidente che $\mathsf{EE}(a, b)$ eredita totalmente i risultati di complessità di $\mathsf{MCD}(a, b)$.[6]

Correttezza. Sappiamo già che d è il massimo comun divisore tra a e b. Per induzione sul numero di chiamate ricorsive mostriamo che x e y si comportano nel modo atteso. Con 0 chiamate ricorsive significa che $b = 0$. Ma allora abbiamo che

$$a1 + b0 = a = \mathsf{MCD}(a, b)$$

[4] Al solito, con $x|y$ (x divide y) si intende che esiste $a \in \mathbb{N}$ tale che $y = xa$, o, equivalentemente, che $y \bmod x = 0$. Con $x \not| y$ si denota che x non divide y.

[5] Verificate, ad esempio, che: $\lim_{n\to\infty} \frac{F_n}{F_{n-1}} = \frac{1+\sqrt{5}}{2}$, da cui si deduce che F_n ha un andamento asintotico come $(\frac{1+\sqrt{5}}{2})^n$ (migliori approssimazioni, tutte basate su tale quantità, legata alla parte aurea del segmento, si trovano con facilità in letteratura).

[6] Facendo uno studio preciso, non asintotico, le costanti ovviamente cambiano leggermente.

Prendiamo il caso induttivo. Sappiamo che $d' = \mathsf{MCD}(a, a \underline{\mathsf{mod}}\, b) = \mathsf{MCD}(a, b)$. Sappiamo per ipotesi induttiva che $bx' + (a \underline{\mathsf{mod}}\, b)y' = d'$. Dobbiamo mostrare che $ay' + b(x' - (a \underline{\mathsf{div}}\, b)y') = d'$. Ricordiamoci che $a = (a \underline{\mathsf{div}}\, b)b + (a \underline{\mathsf{mod}}\, b)$. Allora

$$\begin{array}{lll} ay' + b(x' - (a \underline{\mathsf{div}}\, b)y') & = & \\ (a \underline{\mathsf{div}}\, b)by' + (a \underline{\mathsf{mod}}\, b)y' + bx' - b(a \underline{\mathsf{div}}\, b)y' & = & \\ (a \underline{\mathsf{mod}}\, b)y' + bx' & = & d' \end{array}$$

A cosa può mai servire questo algoritmo? Supponiamo a sia un numero primo, $a > b$ (dunque, in particolare, $\mathsf{MCD}(a, b) = 1$). Eseguendo $\mathsf{EE}(a, b)$ ottengo (in tempo $O(\log b)$) una tripla $\langle 1, x, y\rangle$ tale che $ax + by = 1$. Se ragiono nel campo finito $\mathbb{Z}_a$, y è l'inverso (rispetto a '$\cdot$') di b. Dunque EE mi permette di trovare in modo molto efficiente gli inversi in $\mathbb{Z}_a$.

Concludiamo la sezione con alcuni semplici risultati riguardanti l'aritmetica modulare. Innanzitutto, per ogni $n \in \mathbb{N} \setminus \{0\}$ definiamo la relazione di equivalenza $\equiv_n$ (congruo modulo n) nel seguente modo:

$$a \equiv_n b \text{ se e solo se } (a \underline{\mathsf{mod}}\, n) = (b \underline{\mathsf{mod}}\, n)$$

Dimostriamo dunque che se $ab \equiv_n ac$ e $\mathsf{MCD}(a, n) = 1$, allora $b \equiv_n c$. Sappiamo che esistono x, y per cui $ax + ny = 1$. Ma allora $(b - c)(ax + ny) = b - c$. Dunque $(ab - ac)x + n(b - c)y = b - c$. Per ipotesi $ab \equiv_n ac$, dunque $(ab - ac)$ è multiplo di n. Pertanto, anche $b - c$ lo è. Dunque $b \equiv_n c$.

Come corollario, osserviamo che se $ab \equiv_n 0$ e $\mathsf{MCD}(a, n) = 1$, allora $b \equiv_n 0$.

Vediamo anche che, se p e q sono primi diversi e $a \equiv_p b$ e $a \equiv_q b$ allora $a \equiv_{pq} b$

$a \equiv_p b$ implica che esiste h per cui $a - b = ph$. $a \equiv_q b$ implica che esiste k per cui $a - b = qk$. Pertanto esistono h, k per cui $ph = qk$. Essendo p e q primi diversi, dev'essere che $h = qx$ e $k = px$ per qualche x. Dunque $a - b = pqx$, ovvero $a \equiv_{pq} b$.

4.3 Fermat

Enunciamo e dimostriamo qui il cosiddetto *piccolo teorema di Fermat*, ovvero: se p è primo e $p \not| a$,[7] allora $a^{p-1} \equiv_p 1$. Per dimostrarlo, sia $S = \{1, 2, 3, \ldots, p - 1\}$. Consideriamo una funzione ψ definita come

$$\psi(x) = ax \underline{\mathsf{mod}}\, p$$

Mostriamo che ψ è una funzione biiettiva da S in S.

1) Mostriamo che $\psi(x) \in S$ per ogni $x \in S$. Ovviamente $\psi(x) \in \{0\} \cup S$. Supponiamo $\psi(x) = 0$. Significa che $ax \underline{\mathsf{mod}}\, p = 0$. Ma sappiamo che $\mathsf{MCD}(a, p) = 1$. Allora $a = 0$, assurdo.

2) Mostriamo che ψ è iniettiva. Siano $x, y \in S$ tali che $ax \underline{\mathsf{mod}}\, p = ay \underline{\mathsf{mod}}\, p$, ovvero $ax \equiv_p ay$. Ma sappiamo che $\mathsf{MCD}(a, p) = 1$. Pertanto $x \equiv_p y$. Essendo $x < p$ e $y < p$ ciò significa che $x = y$.

3) Essendo ψ iniettiva da S in S e S insieme finito si ha che ψ biiettiva.

Pertanto $\psi(1), \psi(2), \ldots, \psi(p - 1)$ è una enumerazione di S e dunque

$$\begin{array}{lll} 1 \cdot 2 \cdot \cdots \cdot (p-1) & \equiv_p & \psi(1) \cdot \psi(2) \cdot \cdots \cdot \psi(p-1) \\ & \equiv_p & (a1)(a2) \cdot \cdots \cdot (a(p-1)) \\ & \equiv_p & a^{p-1}(1 \cdot 2 \cdot \cdots \cdot (p-1)) \end{array}$$

Poichè p è primo e maggiore di $1, 2, \ldots, p - 1$ possiamo dividere e dunque $a^{p-1} \equiv_p 1$.

4.4 Eulero

La *funzione di Eulero* Φ viene definita come segue:

$$\Phi(n) = |\{z \in \mathbb{N} : 0 < z < n, \mathsf{MCD}(n, z) = 1\}|$$

Ovviamente, se n è un numero primo, banalmente $\Phi(n) = n - 1$. Supponiamo n non sia primo e $p < n$ sia un primo che divide n. Allora avremo che $\mathsf{MCD}(p, n) = p > 1, \mathsf{MCD}(2p, n) \geq p > 1, \mathsf{MCD}(3p, n) \geq p > 1, \cdots$ Pertanto abbiamo almeno $\frac{n}{p} - 1$ numeri tra 1 e $n - 1$ che non sono contati in $\Phi(n)$. Notiamo

[7] Dunque $a \neq 0$.

che se anche $2|n$ (e $p \neq 2$) allora $2p$ viene contato sia tra quelli da eliminare per 2 che per p. Ragionando un attimo si giunge alla formula:

$$\Phi(n) = n \prod_{p|n, p \text{ primo}} \left(1 - \frac{1}{p}\right)$$

Ad esempio, $\Phi(20) = 20(1 - \frac{1}{2})(1 - \frac{1}{5}) = 20\frac{1}{2}\frac{4}{5} = 8$. Il divisore 2 riduce ad $\frac{1}{2}$ i possibili non divisori (eliminando dunque anche il numero 10, multiplo anche di 5), il divisore 5 riduce di $\frac{4}{5}$ i possibili divisori dei restanti numeri.

Il *Teorema di Eulero* ci dice che: se $\mathsf{MCD}(a, n) = 1$, allora $a^{\Phi(n)} \equiv_n 1$. la sua dimostrazione ricalca quella del piccolo teorema di Fermat vista sopra e la lasciamo per esercizio.

4.5 Operazioni efficienti nei campi finiti

Il Teorema di Eulero ha delle ricadute per l'aritmetica su numeri *grandi* (in senso informale) in modulo. Se vogliamo calcolare ad esempio

$$2^{43210} \mod 101$$

Poichè 101 è primo (proprietà non rara nell'aritmetica in modulo), $\Phi(101) = 100$. Poichè $\mathsf{MCD}(2, 101) = 1$ avremo che

$$2^{43210} \equiv_{101} (2^{100})^{432} \cdot 2^{10} \equiv_{101} 1^{432} 2^{10} \equiv_{101} 1024 \equiv_{101} 14$$

Sempre relativamente al calcolo di un esponente in aritmetica modulare, pur senza usare il Teorema di Eulero, mostriamo come sia possibile farlo in tempo efficiente.

Si voglia ad esempio calcolare $2^{8438} \underline{\mathsf{mod}}\ 789$. Innanzitutto con $\log_2 8438$ divisioni siamo in grado di calcolare la rappresentazione binaria di 8438=8192+128+64+32+16+4+2.

A questo punto calcoliamo le potenze di due fino al valore desiderato:

i	$2^i \underline{\mathsf{mod}}\ 789$
1	2
2	$2^2 = 4$
4	$4^2 = 16$
8	$16^2 = 256$
16	$256^2 = 65536 \equiv_{789} 49$
32	$49^2 = 2401 \equiv_{789} 34$
64	$34^2 = 1156 \equiv_{789} 367$
128	$367^2 = 134689 \equiv_{789} 559$
256	$559^2 = 312481 \equiv_{789} 37$
512	$37^2 = 1369 \equiv_{789} 580$
1024	$580^2 = 336400 \equiv_{789} 286$
2048	$286^2 = 81796 \equiv_{789} 529$
4096	$529^2 = 279841 \equiv_{789} 535$
8192	$535^2 = 286225 \equiv_{789} 607$

Ad ogni passo facciamo il quadrato del numero ottenuto al passo precedente e lo modularizziamo, mantenendo sempre numeri "piccoli" (possiamo pertanto immaginare operazioni su interi in tempo costante). A questo punto si tratta di fare altre al più $\log_2 8438$ moltiplicazioni e divisioni:

$$\begin{aligned} 2^{8438} &= (((((2^{8192} 2^{128} \underline{\mathsf{mod}}\ 789) 2^{64} \underline{\mathsf{mod}}\ 789) 2^{32} \underline{\mathsf{mod}}\ 789) 2^{16} \underline{\mathsf{mod}}\ 789) 2^4 \underline{\mathsf{mod}}\ 789) 2^2 \underline{\mathsf{mod}}\ 789 \\ &= (((((607 \cdot 559 \underline{\mathsf{mod}}\ 789) 367 \underline{\mathsf{mod}}\ 789) 34 \underline{\mathsf{mod}}\ 789) 49 \underline{\mathsf{mod}}\ 789) 16 \underline{\mathsf{mod}}\ 789) 4 \underline{\mathsf{mod}}\ 789 \\ &= 109 \end{aligned}$$

Pertanto, l'esponenziale finito a^n si calcola in tempo proporzionale a $\log_2 n$.