

CRI

lezione del 30/09

---

Generatori di numeri  
pseudo casuali

---

---

---

---



## Sorgente casuale binaria

genera una sequenza di bit tc.

- 1)  $P(0) = P(1) = \frac{1}{2}$
- 2) la generazione di un bit è indipendente da quelli degli altri bit

→  $P(0) > 0$     $P(1) > 0$    immutabile durante il processo

per ipotesi si è  $P(0) > P(1)$

~~0 1 | 1 0 | 0 1 | 0 1 | 0 1 | 1 0 | 0 1 | 0 1 | 0 1 | 0 1~~

$0 1 \rightarrow 0$

$1 0 \rightarrow 1$

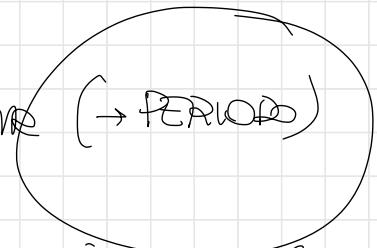
$0 1 1 0 0 1 0$

# Generazione di numeri pseudo casuali

Idee si genera la casualezza mediante un algoritmo, cercandola all'interno di processi matematici

INPUT: sequenze brevi SENS

OUTPUT: flusso di bit arbitrariamente lungo,  
che all'interno contiene una sottosequenza  
che si ripete



Un generatore è buon negliari quanto più lunga  
è il suo periodo.

S : sequenza  $|S| = s$  bit

Il generatore genera al più

$$2^s \ll 2^n$$

$2^s$  figure nere diverse

$$(s < n)$$

## Generatore lineare

$$x_i = (a \cdot x_{i-1} + b) \bmod m$$

$a, b, m$  parametri  
del generatore

$$x_0, x_1 = (ax_0 + b) \bmod m, \quad a, b, m \in \mathbb{N}$$

$$x_2 = (ax_1 + b) \bmod m, \dots$$

il periodo è  $\leq m$

se i periodi sono tutti le stesse, il generatore  
modello una permutazione degli interi

$$[0, m-1]$$

## CRITERI

$$\text{gcd}(b, m) = 1$$

$(a-1)$  deve essere divisibile per ogni fattore primo  
di  $m$

~~$(a-1) \mid m \Rightarrow 4 \mid a-1$~~

$\rightsquigarrow$  per sequenze binarie

$\frac{x_i}{m} \rightarrow$  si prende le parti ~~dopo~~ delle prime cifre decimali -

### Generatore polinomiale

$$x_i = (\underbrace{a_1 x_{i-1}^t + a_2 x_{i-1}^{t-1} + \dots + a_t x_{i-1}}_{\underbrace{\hspace{10em}}_{\text{LINEARE}}} + \dots + a_{t+1}) \mod m$$

LINEARE }  $a = 3141592653$   
 $b = 2718281829$

$$m = 2^{32}$$

## Test statistici

- 1) test di frequenze
- 2) ~~the~~ poker test
- 3) test di setteconclavine
- 4) non test

X ~~to~~ le applicazioni cromografiche riduttive  
il test di grossmo bit

## TEST di PROSSIMO BIT

Un generatore binario supera il test del prossimo bit se l'algoritmo polinomiale è in grado di prevedere l' $(i+1)$ -esimo bit generato a partire dalla conoscenza degli  $i$ -bit precedentemente generati con probabilità  $> \frac{1}{2}$ .

Un generatore è critograficamente sicuro se supera il test del prossimo bit

## Generazioni basati su fermioni one-way

→ è computazionalmente facile calcolare  $y = f(x)$

$$x \rightarrow y = f(x) \quad \text{tempo polinomiale}$$

→ è computazionalmente difficile calcolare  $x = f^{-1}(y)$

↳ conosciamo solo alg. di Carlo  
esponenziale

$f$ : one-way

$$x_0 \quad x_1 = f(x_0) \quad x_2 = f(x_1) = f(f(x_0)) = f^{(2)}(x_0)$$

seme

$$\dots \quad x_i = f(x_{i-1}) = f^{(i)}(x_0) \quad \dots$$

$$x_0 \ x_1 \ x_2, \dots, x_i \ x_{i+1}, \dots$$

Si genera la sequenza, e si consuma la sequenza  
in ordine inverso

$x_n, x_{n-1}, \dots, x_{i+1}, x_i, x_{i-1}, \dots, x_2, x_1, x_0$

Se  $x_{i+1}$  è esp. detti cele processore  $x_i$ .

$$x_{i+1} = f(x_i)$$



$$x_i = f^{-1}(x_{i+1})$$

## Generatore binario entropico con rullo

"predicati hard-core" delle fermioni ~~per~~ one-way

$b(x)$  è ~~un~~ un predicato hard-core di un fermione one-way  $f(x)$  se

$b(x)$  è facile da calcolare conoscendo  $x$

$b(x)$  è difficile da prendere con probabilità  $> \frac{1}{2}$  se si conosce  $f(x)$

$f$ : elemento di questo insieme per  
memoro compito

$b$ : può-

$$x = 10$$

mod 77

$$f(x) = 10^2 \text{ mod } 77 = 23$$

Generatore BBS (Blum, Blum, Shub, 1986)

1993 primo tuning

o' antropologamente sicuro

$n = p \times q$   $p < q$  numeri primi (grandi)

$$p \bmod 4 = 3$$

$$q \bmod 4 = 3$$

$2 \mid p_4$  e  $2 \mid q_4$  siano primi fra loro

y coprimo con n

calcoliamo  $x_0 = y^2 \bmod n$   
sia

generiamo una successione di  $m \leq n$  interi

$$x_i = (x_{i-1})^2 \bmod n \quad i \geq 1$$

$x_0 \ x_1 \ x_2 \ \dots \ x_{i-1} \ x_i \ x_{i+1} \ \dots \ x_m$

$$b_i = 1 \iff x_{m-i} \text{ è } \text{defini}$$

$b_m \ b_{m-1} \ \dots \ b_2 \ b_0$



ESEMPIO

$$P = 11$$

$$Q = 19$$

$$n = \underline{209}$$

$$y = 30$$

$$\text{gcd}(30, 209) = 1$$

$$x_0 = 30^2 \bmod 209 = 64$$

$$p \bmod 4 = \cancel{11} = 3 \bmod 4 = 3$$

~~$$q \bmod 4 = 19 \bmod 4 = 3$$~~

$$2 \left\lfloor \frac{p}{q} \right\rfloor + 1 \rightarrow \cancel{7}$$

$$2 \left\lfloor \frac{q}{p} \right\rfloor + 1 \rightarrow 9$$

$$x_0 \quad x_1 \quad x_2 \quad x_3 \quad - \quad - \quad -$$

$$64 \quad 125$$

$$159 \quad 201$$

$$\odot$$

$$1$$

$$1$$

$$1$$



# Generazione di numeri pseudocasuali basati su cipheri simmetrici

- ciphero simmetrico (DES, 3DES, AES)

$r = \#$  bit delle parole prodotte ( $r = 64$ , 128, 256)  
DES, AES

$S = \#$  parola di  $r$  bit

$M = \#$  parole (sequenze) di  $r$  bit

$k = \#$  chiavi segrete del ciphero

## Generatore ( $\Sigma$ , $\underline{m}$ )

$d$  = rappresentazione su  $r$  bit di data e ora

$$y = G(d, k)$$

$$z = s$$

for ( $i = 1$ ;  $i \leq m$ ;  $i++$ )

$$x_i = G(y \oplus z, k);$$

$$z = G(y \oplus x_i, k);$$

Quindi  $x_i$  sì esiste;

$y$

$\oplus$  : XOR  
bit a bit

$$\begin{array}{r} 1001 \\ 1100 \\ \hline 0101 \end{array}$$