Cursul 8

Generatorul Mersenne–Twister

Algoritmii de generare de numere pseudo–aleatoare uniform distribuite pe $[0, 1)$, de ultimă generaţie, sunt bazaţi pe recurenţe liniare modulo 2. Aceşti algoritmi se bazează pe operaţii cu stringuri de biţi.

Se consideră corpul finit $\mathbb{Z}_2 = \{0, 1\}$, al claselor de resturi modulo 2, relativ la operaţiile de adunare şi înmulţire modulo 2. Adunarea modulo 2 se numeşte `exclusive or` şi se notează `XOR` (în `C` operaţia pe biţi `XOR` se notează `^` , iar în criptografie $\oplus$):

XOR	0	1
0	0	1
1	1	0

$\mathbb{Z}_2$ fiind corp, $\mathbb{Z}_2^n$ are structură de spaţiu vectorial peste corpul $\mathbb{Z}_2$, tot aşa cum $\mathbb{R}^n$ are structură de spaţiu vectorial peste $\mathbb{R}$. Deoarece

$$\mathbb{Z}_2^n = \underbrace{\mathbb{Z}_2 \times \mathbb{Z}_2 \times \cdots \times Z_2}_{n},$$

cardinalul mulţimii $\mathbb{Z}_2^n$ este 2^n.

O matrice A de tip $n \times n$ cu elemente în $\mathbb{Z}_2$ defineşte un operator liniar pe $\mathbb{Z}_2^n$, $A : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$. A asociază stringului de biţi $b_{n-1}b_{n-2} \ldots b_1b_0$, stringul $c_{n-1}c_{n-2} \ldots c_1c_0$ şi relaţia dintre cele două stringuri exprimată matriceal este:

$$\left[\begin{array}{cccc} b_{n-1} & \ldots & b_1 & b_0 \end{array} \right] \cdot A = \left[\begin{array}{cccc} c_{n-1} & \ldots & c_1 & c_0 \end{array} \right].$$

Prezentăm în continuare un algoritm de generare de numere pseudo–aleatoare pe w biţi, unde w este lungimea unui cuvânt (w=32 pe sistemele pe 32 de biţi, respectiv w=64, pe cele pe 64 de biţi), datorat lui M. Matsumoto şi T. Nishimura. Şirul de numere este definit de o transformare liniară particulară, ce acţionează asupra unui spaţiu $\mathbb{Z}_2^N$, cu N ales într-un mod special.

Notaţii:

- w, lungimea unui cuvânt pe un calculator (w=32, 64);
- n, întreg pozitiv fixat;
- r, întreg cu $0 \leq r \leq w - 1$, ce indică punctul de separare al unui cuvânt;
- $\mathbf{x} = (b_{w-1}b_{w-2} \ldots b_r | b_{r-1} \ldots b_0) \in \mathbb{Z}_2^w$;

Vectorului $\mathbf{x}$ i se asociază

$$\mathbf{x}^u = (b_{w-1}b_{w-2}\dots b_r 0\dots 0),\ \mathbf{x}^l = (0\dots 0 b_{r-1}\dots b_0);$$

- m, întreg pozitiv fixat, $1 \le m \le n$;
- $k = 0, 1, 2\dots$, ordinul recurenţei;

Algoritmul este definit de următoarea recurenţă liniară:

$$\mathbf{x}_{n+k} = \mathbf{x}_{m+k} \oplus (\mathbf{x}_k^u | \mathbf{x}_{k+1}^l)A, \quad k = 0, 1, 2, \dots \tag{8.1}$$

unde matricea:

$$A = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & 1 \\ a_{w-1} & a_{w-2} & a_{w-3} & \dots & a_0 \end{pmatrix} \tag{8.2}$$

este asociată unui vector (string) cuvânt $\mathbf{a} = (a_{w-1}, a_{w-2}, \dots, a_1, a_0) \in \mathbb{Z}_2^w$;

- $(\mathbf{x}_k^u | \mathbf{x}_{k+1}^l)$ notează concatenarea substringurilor lui $\mathbf{x}_k^u$, $\mathbf{x}_{k+1}^l$, formate respectiv din cei mai reprezentativi $w-r$ biţi din $\mathbf{x}_k$ şi cei mai puţin reprezentativi r biţi din $\mathbf{x}_{k+1}$. Mai precis, dacă $\mathbf{x}_k = (\mathtt{b_{w-1}\dots b_r b_{r-1}\dots b_0})$, iar $\mathbf{x}_{k+1} = (\mathtt{c_{w-1}\dots c_r c_{r-1}\dots c_0})$, atunci avem:

$$\begin{aligned} &(\mathbf{x}_{\mathtt{k}}^{\mathtt{u}} | \mathbf{x}_{\mathtt{k+1}}^{\mathtt{l}}) = \\ &((\mathtt{b_{w-1}\dots b_r b_{r-1}\dots b_0})\ \mathtt{AND}\ (\mathtt{1\dots 1}\underbrace{\mathtt{0,\dots 0}}_{\mathtt{r}}))\ \mathtt{OR} \\ &((\mathtt{c_{w-1}\dots c_r c_{r-1}\dots c_0})\ \mathtt{AND}\ (\mathtt{0\dots 0}\underbrace{\mathtt{1\dots 1}}_{\mathtt{r}})). \end{aligned}$$

Evident că $(\mathbf{x}_k^u | \mathbf{x}_{k+1}^l)$ este un w–string.

- Operatorul $\oplus$ din (8.1) este adunarea modulo 2.

Detalierea relaţiei (8.1):

Se porneşte cu condiţiile iniţiale $\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_{n-1}$, ce sunt n numere întregi fără semn, pe w–biţi, nu toate nule.

Relaţia de recurenţă (8.1) calculează pentru $k = 0, 1, \dots, n-1$, un nou set de n numere:

$$\begin{aligned} \mathbf{x}_n &= \mathbf{x}_m \oplus (\mathbf{x}_0^u | \mathbf{x}_1^l)A, \\ \mathbf{x}_{n+1} &= \mathbf{x}_{m+1} \oplus (\mathbf{x}_1^u | \mathbf{x}_2^l)A, \\ &\vdots \\ \mathbf{x}_{2n-2} &= \mathbf{x}_{m+n-2} \oplus (\mathbf{x}_{n-2}^u | \mathbf{x}_{n-1}^l)A, \\ \mathbf{x}_{2n-1} &= \mathbf{x}_{m+n-1} \oplus (\mathbf{x}_{n-1}^u | \mathbf{x}_n^l)A. \end{aligned} \tag{8.3}$$

Particularizând aceste relaţii pentru $n = 5, m = 3$:

$$\begin{aligned} \mathbf{x}_5 &= \mathbf{x}_3 \oplus (\mathbf{x}_0^u | \mathbf{x}_1^l)A, \\ \mathbf{x}_6 &= \mathbf{x}_4 \oplus (\mathbf{x}_1^u | \mathbf{x}_2^l)A, \\ \mathbf{x}_7 &= \mathbf{x}_5 \oplus (\mathbf{x}_2^u | \mathbf{x}_3^l)A, \\ \mathbf{x}_8 &= \mathbf{x}_6 \oplus (\mathbf{x}_3^u | \mathbf{x}_4^l)A, \\ \mathbf{x}_9 &= \mathbf{x}_7 \oplus (\mathbf{x}_4^u | \mathbf{x}_5^l)A, \end{aligned} \tag{8.4}$$

observăm că se poate folosi ca memorie de lucru doar zona alocată pentru

$$\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_{n-1},\ n = 5,$$

făcând atribuirile din (8.5), care sunt cele din (8.4), în care fiecare indice mai mare sau egal cu 5 se înlocuieşte cu restul împărţirii sale la 5:

$$\begin{array}{rcl} \mathbf{x}_0 & = & \mathbf{x}_3 \oplus (\mathbf{x}_0^u|\mathbf{x}_1^l)A, \\ \mathbf{x}_1 & = & \mathbf{x}_4 \oplus (\mathbf{x}_1^u|\mathbf{x}_2^l)A, \\ \mathbf{x}_2 & = & \mathbf{x}_0 \oplus (\mathbf{x}_2^u|\mathbf{x}_3^l)A, \\ \mathbf{x}_3 & = & \mathbf{x}_1 \oplus (\mathbf{x}_3^u|\mathbf{x}_4^l)A, \\ \mathbf{x}_4 & = & \mathbf{x}_2 \oplus (\mathbf{x}_4^u|\mathbf{x}_0^l)A. \end{array} \tag{8.5}$$

În relaţiile (8.3) indicii se calculează modulo n, adică pentru $k \in \{0, 1, \ldots, n-m-1\}$ avem:

$$\mathbf{x}_{n+k\,(\mathrm{mod}\,n)} = \mathbf{x}_{m+k} \oplus (\mathbf{x}_k^u|\mathbf{x}_{k+1}^l)A,$$

pentru $k \in \{n-m, \ldots, n-2\}$:

$$\mathbf{x}_{n+k\,(\mathrm{mod}\,n)} = \mathbf{x}_{m-n+k} \oplus (\mathbf{x}_k^u|\mathbf{x}_{k+1}^l)A,$$

iar pentru $k = n-1$:

$$\mathbf{x}_{2n-1\,(\mathrm{mod}\,n)} = \mathbf{x}_{n-1} = \mathbf{x}_{m-1} \oplus (\mathbf{x}_{n-1}^u|\mathbf{x}_0^l)A.$$

Pentru a deduce modul de transformare a înmulţirii la dreapta a vectorului linie

$$\mathbf{x} = [b_{w-1}, \ldots, b_1, b_0]$$

cu matricea A, în operaţii pe biţi, în care să fie implicaţi $\mathbf{x}$ şi $\mathbf{a}$, luăm cazul particular $w = 4$ şi avem:

$$[\ b_3 \quad b_2 \quad b_1 \quad b_0\ ] \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ a_3 & a_2 & a_1 & a_0 \end{bmatrix} = [\ b_0a_3 \quad b_3 + b_0a_2 \quad b_2 + b_0a_1 \quad b_1 + a_0b_0\ ]. \tag{8.6}$$

Analizând produsul, constatăm că în general avem pentru $\mathbf{x} = (b_{w-1} \ldots b_2b_1b_0)$:

$$\mathbf{x} \cdot A = \begin{cases} \mathbf{x} >> 1 = (0b_{w-1} \ldots b_1), & \text{dacă } b_0 = 0, \\ \mathbf{x} >> 1 \oplus a = (0b_{w-1} \ldots b_2b_1) \oplus (a_{w-1}a_{w-2} \ldots a_1a_0), & \text{dacă } b_0 = 1. \end{cases} \tag{8.7}$$

Precizăm că `>>` reprezintă operatorul pe biţi de deplasare la dreapta.

Fiecărui vector $\mathbf{x} \in \mathbb{Z}_2^w$ i se aplică un şir de transformări, numite transformări de temperare, care îmbunătăţesc proprietăţile statistice (echidistribuţia) ale şirului generat. Aceste transformări sunt:

$$\mathbf{y} = \mathbf{x} \oplus (\mathbf{x} >> u) \tag{8.8}$$

$$\mathbf{y} = \mathbf{y} \oplus ((\mathbf{y} << s)\ \texttt{AND b}) \tag{8.9}$$

$$\mathbf{y} = \mathbf{y} \oplus ((\mathbf{y} << t)\ \texttt{AND c}) \tag{8.10}$$

$$\mathbf{z} = \mathbf{y} \oplus (\mathbf{y} >> v), \tag{8.11}$$

unde `s,t, u,v` sunt numere întregi fără semn, iar `b, c` sunt măşti adecvate de lungime w biţi.

După ce s-au aplicat tranformările de temperare, numărului întreg $\mathbf{z}$ i se asociază un număr din $[0, 1)$, prin transformarea de ieşire $g(\mathbf{z}) = \mathbf{z}/M$, unde M este cel mai mare număr întreg fără semn (în `C`, pentru $w = 32$, acest număr este `0xffffffff`).

În concluzie, pentru a calcula $N > n$ numere întregi fără semn, de w biţi, cu ajutorul relaţiei de recurenţă (8.1), pe baza a n numere iniţiale, avem nevoie doar de un domeniu de lucru constând dintr-un tablou `x[n]` de n numere întregi de tip `unsigned long`.

Se poate demonstra că relaţiile (8.3) se pot scrie în forma:

$$s^{\ell} = s^{\ell-1}M, \tag{8.12}$$

unde M este o matrice cu elemente din Z_2, de tip $(nw - r) \times (nw - r)$, iar s^{ℓ} este un vector linie de $nw - r$ biţi. Perioada maximă a generatorului se realizează dacă polinomul caracteristic al matricei M este un polinom primitiv cu coeficienţi din $\mathbb{Z}_2$.

Autorii algoritmului au arătat că dacă n, w, r sunt astfel încât $2^{nw-r} - 1$ este un număr prim, atunci polinomul asociat matricei M a generatorului (8.1) este polinom primitiv.

Matsumoto şi Nishimura au dat în lucrarea respectivă şi parametrii ce definesc prin metoda prezentată mai sus cel mai bun generator de numere pseudo–aleatoare uniform distribuite pe $[0, 1)$ existent la ora actuală:

- n=624; m=397
- vectorul `a`: `0x9908b0df`
- masca `umsk` pentru calculul lui $\mathbf{x}_k^u$: `0x80000000`
- masca `lmsk` pentru calculul lui $\mathbf{x}_{k+1}^l$: `0x7fffffff`, deci $r = 31$
- parametrii de temperare:
  `b`: `0x9d2c5680`
  `c`: `0xefc60000`
  `u`: `11, adică se calculează (y>>11)`
  `s`: `7`
  `t`: `15`
  `v`: `18`.

Generatorul definit de aceşti parametri se numeşte generatorul Mersenne-Twister. El are următoarele proprietăţi:

- Perioada maximă a şirului generat este de $2^{nw-r} - 1$. Deci pentru $n = 624$, $w = 32$ şi $r = 31$, perioada este $2^{19937} - 1$;
- A trecut teste de k–uniformitate pentru orice $k \leq 623$, adică s-a studiat distribuţia punctelor în hypercuburi de dimensiuni de la 2 până la 623 şi s-a constatat că nu are proprietăţi de regularitate ca generatorul liniar congruenţial;
- Necesită $624 \times 4 = 2496$ bytes ca memorie de lucru pe un calculator pe 32 de biţi.